

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ

КАФЕДРА СИСТЕМНОГО ПРОГРАМУВАННЯ І
СПЕЦІАЛІЗОВАНИХ КОМП'ЮТЕРНИХ СИСТЕМ

«На правах рукопису»
УДК 004.021

«До захисту допущено»

Завідувач кафедри СПСКС

Віталій РОМАНКЕВИЧ
(підпис) (ім'я, прізвище)
“ ____ ” _____ 2020р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 123 Комп'ютерна інженерія

на тему: Адаптивний алгоритм керування роботою двигуна автомобіля ____

Виконав: студент II курсу, групи КВ-92мп

Побігай Антон Олександрович _____
(прізвище, ім'я, по батькові) (підпис)

Науковий керівник к.т.н., доцент Тарасенко-Клятченко О. В. _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Рецензент _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Консультант з нормоконтролю доцент, с.н.с., к.т.н. Юлія БОЯРІНОВА _____
(посада, науковий ступінь, вчене звання, прізвище та ініціали) (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ – 2020 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет прикладної математики

Кафедра системного програмування і спеціалізованих комп'ютерних систем

Рівень вищої освіти – другий (магістерський)

за освітньо-професійною програмою

Спеціальність 123 Комп'ютерна інженерія (Системне програмування)

ЗАТВЕРДЖУЮ

Завідувач кафедри СПСКС

Віталій РОМАНКЕВИЧ

(підпис) (ім'я, прізвище)

«__» _____ 2020 р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Побігай Антону Олександровичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Адаптивний алгоритм керування роботою двигуна автомобіля»

науковий керівник дисертації к.т.н., доцент Тарасенко-Клятченко О. В.,
затверджені наказом по університету від «12» листопада 2020 р. № 3298-С

2. Термін подання студентом дисертації 10 грудня 2020 р. _____

3. Об'єкт дослідження: налаштування параметрів роботи двигуна автомобіля в реальному часі _____

4. Предмет дослідження: програмне забезпечення ЕБУ двигуна автомобіля

5. Перелік завдань, які потрібно розробити: проаналізувати існуючі рішення, вивчення методів підняття потужності, розробити адаптивний алгоритм, протестувати роботу розробленого алгоритму, навести приклад роботи моделі та обґрунтувати результати. _____

6. Перелік ілюстративного матеріалу: презентація _____

7. Перелік публікацій: _____

8. Дата видачі завдання 5 вересня 2019 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за тематикою дисертації	11.10.2019	
2.	Підготовка матеріалів першого розділу магістерської дисертації	16.09.2020	
3.	Підготовка матеріалів другого розділу магістерської дисертації	22.09.2020	
4.	Підготовка матеріалів третього розділу магістерської дисертації	26.09.2020	
5.	Підготовка матеріалів четвертого розділу магістерської дисертації	30.09.2020	
6.	Підготовка матеріалів п'ятого розділу магістерської дисертації	05.10.2020	
7.	Розробка адаптивного алгоритму	09.10.2020	
8.	Тестування розробленого алгоритму	15.10.2020	
9.	Оформлення документації магістерської дисертації	21.11.2020	
10.	Попередній розгляд магістерської дисертації на кафедрі	26.11.2020	

Студент _____
(підпис)

Побігай А.О.

Науковий керівник дисертації _____
(підпис)

Тарасенко-Клятченко О.В.

РЕФЕРАТ

Актуальність теми.

Бажання людини пересуватися швидше було присутнє нам з давніх давен, але не завжди це було безпечно та надійно. При досягненні цієї мети необхідно враховувати дуже багато дрібниць, оскільки чим вища потужність – тим більший ризик для механізму та для водія. У сучасному світі, коли управління двигуна автомобіля виконується переважно електронікою. Головна перевага такого виконання – більший діапазон налаштувань та засобів діагностики автомобіля. Це дозволяє виробнику продавати один і той самий двигун в різних автомобілях, в різних комплектаціях та за різну ціну. Процес глобального впровадження електронних рішень все більше витісняє механічне керування, що позитивно впливає на стабільність та надійність агрегатів, але негативно відображається на поведінку автомобіля у випадку відмови одного із сенсорів.

Об'єктом дослідження є процес налаштування параметрів роботи двигуна автомобіля в реальному часі.

Предметом дослідження є програмне забезпечення ЕБУ двигуна автомобіля.

Мета роботи – підвищення ефективності (збільшення потужності та зменшення витрат палива автомобілем) за рахунок створення нових програмно-апаратних засобів для реалізації адаптивного алгоритму керування двигуном автомобіля і, робота яких, буде заснована на аналізі характеристик навколишнього середовища та внутрішніх параметрів двигуна, трансмісії й інших вузлів автомобіля.

Методи дослідження. Найважливішим методом дослідження є діагностика роботи двигуна, аналіз отриманих даних в процесі діагностики та власне розробка тому, що магістерська дисертація присвячена вивченню великих об'ємів здобутих знань у питаннях підвищення ефективності роботи двигуна автомобіля, шляхом оптимізації роботи механізму, що керується електронно, їх обробці та реалізації способів вдосконалення. Також було використано метод ігнорування, що дозволяє програмному забезпеченню не звертати увагу на певні механічні несправності, що можуть заважати якісній роботі ДВЗ.

Наукова новизна роботи полягає в тому, що має найвищий рівень адаптації алгоритму керування роботою двигуна автомобіля, а також алгоритм має більший функціонал в порівнянні з вже існуючими програмними рішеннями, завдяки унікальній адаптивній схемі управління.

Практична цінність полягає у вдосконаленні роботи механізмів двигуна, керування яких відбувається електронікою за для покращення потужності, динамічних та економічних характеристик двигуна автомобіля.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п'ятих розділів, висновків та додатків.

У вступі надано загальний опис роботи існуючих рішень, проведений аналіз переваг та недоліків, пояснена необхідність створення адаптивного алгоритму, сформульована мета роботи.

У першому розділі описується основні принципи збільшення потужності; обрано дослідну модель та обґрунтовано вибір; досліджено цілі та ризики підняття потужності; проаналізовано існуючі рішення; розглянуто недоліки існуючих рішень.

У другому розділі описуються методи розробки програмного забезпечення для двигуна автомобіля та детально проаналізовано кожний із методів.

У третьому розділі розглянуто роль інформаційних технологій в автомобілі; описано використання технології Autosar; проаналізовано процес віддаленого програмного забезпечення; описана побудова функцій платформи автомобіля та розробка додатків; проведена експериментальна перевірка.

У четвертому розділі описується основні принципи адаптивного алгоритму; розроблено принцип роботи адаптивного алгоритму під великим та малим навантаженням; проаналізована робота за умови негативних проявів.

У п'ятому розділі проведено тестування адаптивного алгоритму на реальних автомобілях; досліджено переваги та недоліки.

У висновках представлені результати та порівняння з іншими існуючими програмними засобами підвищення потужності.

Ключові слова адаптивний алгоритм, ЕБУ, CAN-шина, процес запису та обміну інформації.

Abstract

Actuality of theme.

Human's desire to move faster has been with us since time immemorial but it has not always been safe enough. When achieving this goal it is necessary to take care about a lot of little things, because the higher the power - the greater risk for the mechanism and for the driver. In today's world when the car engine is controlled mainly by electronics. The main advantage of this design is a wider range of settings and tools for car diagnostics. This allows the manufacturer to sell the same engine in different cars in different configurations and at different prices. The process of global implementation of electronic solutions is increasingly displacing mechanical control which has a positive effect on the stability and reliability of the units but negatively effects the behavior of the car in the event of failure of one of the sensors.

Object of study is a process of adjusting the parameters of the car engine in real time.

Subject of study is a software of the ECU of car engine.

The purpose of the work is to increase efficiency (increase power and reduce fuel consumption by the car) by creating new software and hardware to implement an adaptive algorithm for controlling the car engine and whose work will be based on analysis of environmental characteristics and internal diameters of the engine, transmission and other components.

Research methods. The most important research method is engine diagnostics and analysis of data obtained in the diagnostic process and the actual development because the master's dissertation is devoted to the study of large amounts of knowledge in improving the efficiency of car engine by optimizing electronically controlled mechanism their processing and implementation of ways to improve. An ignore method was also used which allows the software to ignore certain mechanical faults that may interfere with the quality of engine.

The scientific novelty of the work is that it has the highest level of adaptation of the car engine control algorithm, and the algorithm has more functionality compared to existing software solutions, thanks to a unique adaptive control scheme.

The practical value is in improving the operation of the engine mechanisms which are controlled electronically to adjust the power, dynamic and economic characteristics of the car engine.

Structure and scope of work. The master's thesis consists of an introduction, five chapters, conclusions and appendices.

The introduction presents a general description of the existing solutions, an analysis of the advantages and disadvantages, explains the need to create an adaptive algorithm, formulates the purpose of the work.

The first section describes the basic principles of increasing power; the model is chosen experimentally and the choice is substantiated; the goals and risks of power increase are investigated; existing solutions are analyzed; the shortcomings of existing solutions are considered.

The second section describes the methods of software development for the car engine and analyzes in detail each of the methods.

The third section inspected the role of information technology in the car; describes the use of Autosar technology; the process of remote software is analyzed; described the construction of the functions of the car platform and application development; an experimental test was performed.

The fourth section describes the basic principles of the adaptive algorithm; the principle of operation of the adaptive algorithm under high and low load is developed; analyzed work under conditions of negative manifestations.

The fifth section tests the adaptive algorithm on real cars; the advantages and disadvantages are investigated.

In the conclusions present the results and comparisons with other existing software tools to increase capacity.

Keywords adaptive algorithm, ECU, CAN-bus, process of recording and information exchange

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ	3
ВСТУП	4
1. ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БЛОКІВ КЕРУВАННЯ РОБОТОЮ ДВИГУНА АВТОМОБІЛЯ	6
1.1 Основні передумови для збільшення потужності двигуна автомобіля	6
1.2 Вибір дослідної моделі та обґрунтування вибору	8
1.3 Цілі та ризики підняття потужності	10
1.4 Особливості програмного забезпечення для блоків керування.....	13
двигуном автомобіля	13
1.5 Недоліки існуючих рішень.....	15
Висновки до розділу 1	18
1. СПОСОБИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ДВИГУНА	19
2.1 Способи розробки програмного забезпечення для двигуна автомобіля	19
2.2 Підвищення потужності за допомогою таблиці коефіцієнтів	19
2.3 Підвищення потужності шляхом описування моделі	21
2.4 Підвищення потужності за допомогою лише з одного параметру	23
2.5 Індивідуальне налаштування на основі показників лямбда-зонду	24
Висновки до розділу 2	27
2. ПРОГРАМНІ ЗАСОБИ ДЛЯ ВНЕСЕННЯ ЗМІН В ПАРАМЕТРИ КЕРУВАННЯ АВТОМОБІЛЕМ.....	28
3.1 Особливості застосування інформаційних технологій в автомобілі	28
3.2 Використання технології Autosar	30
3.3 Дистанційне оновлення програмного забезпечення.....	39
3.4 Дослідження систем безпеки електронного модуля керування	41

3.5 Реалізація функцій платформи автомобіля та розробка додатків	46
3.6 Експериментальна перевірка	51
Висновки до розділу 3	52
4. РЕАЛІЗАЦІЯ АДАПТИВНОГО АЛГОРИТМА	53
4.1 Основні принципи адаптивного алгоритма	53
4.2 Опис роботи адаптивного алгоритму під великим навантаженням.....	63
4.3 Опис роботи адаптивного алгоритму під малим навантаженням	65
4.4 Робота з негативними проявами	66
Висновки до розділу 4	68
5. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ РОБОТИ РОЗРОБЛЕНИХ ЗАСОБІВ, ЩО РЕАЛІЗУЮТЬ АДАПТИВНИЙ АЛГОРИТМ	69
5.1 Результати роботи адаптивного алгоритму.....	69
Висновки до розділу 5	79
ВИСНОВКИ	80
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	82

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ЕБУ/ECU (electronic control unit) – електронний блок управління.

TDi (turbo diesel injections) – турбо-дизельний двигун з безпосереднім впорскуванням пального.

ESC (electronic stability control) – електронна система динамічної стабілізації.

BEV (battery electric vehicle) – електромобілі з батареєю.

API (application programming interface) – набір визначень підпрограм, протоколів взаємодії та засобів для створення програмного забезпечення.

OEM (Original equipment manufacturer) – виробники оригінального обладнання.

VMM (virtual machine monitor) – монітор віртуальної машини.

HUD (head-up display) – є частиною графічного інтерфейсу користувача, яка дозволяє отримати різноманітну ігрову інформацію не викликаючи додаткові меню.

AES (Advanced Encryption Standard) – симетричний алгоритм блочного шифрування.

RSA (аббревіатура від прізвищ Rivest, Shamir та Adleman) – криптографічний алгоритм з відкритим ключем

LAN (local area network) – Локальна комп'ютерна мережа є об'єднанням певного числа комп'ютерів на відносно невеликій території

CAN (controller area network) – є послідовною шиною, що підтримує одночасну роботу багатьох ведучих пристроїв.

ВСТУП

Метод підвищення потужності двигуна автомобіля шляхом перепрограмування електронного блоку управління існує вже більше 40 років. З кожним роком виробники електронних модулів керування удосконалюють свої продукти, щоб надати виробникам автомобілів більше систем для регулювання та контролю за станом всіх механічних та електронних вузлів автомобіля. З кожною новою версією кількість регулювань та сенсорів для діагностики зростала, з ними зростала і швидкість обміну та обробки інформації.

Автомобільні виробники почали користуватися можливістю маніпулювати програмним забезпеченням автомобіля в маркетингових цілях, а саме випускати велику кількість автомобілів з однаковими двигунами, однаковими запасними частинами, але з різною потужністю, і продавати їх за різною ціною.

Загально відомі компанії підходять до процесу створення програмного забезпечення більш відповідально. Такі організації, як Revo Technick, купують автомобіль з двигуном, для якого слід розробити програмне забезпечення з більшою потужністю, і на розробку витрачають від 6-ти місяців до 2-х років. Куплений організацією автомобіль стає особистим автомобілем інженера. Після тестування автомобіля в умовах міста, траси та на динамометричному стенді автомобіль відправляють до ПАР. У південній Африці в умовах жаркого клімату програмне забезпечення доналаштовують до умов жаркого клімату. Після вдалих налаштувань автомобіль відправляють до Канади. ЕБУ налаштовується, щоб автомобіль демонстрував найкращі результати в країнах з холодними кліматичними умовами. Також, автомобіль доставляють до країн в яких низька якість пального для адаптації прошивки під експлуатацію в цих країнах. Весь світ розділяють на регіони і під кожний регіон є своя версія програмного забезпечення. Програмний продукт цієї компанії взагалі не шкодить ресурсу та надійності автомобіля, і при цьому має більшу кількість систем безпеки, ніж заводські версії програми управління.

Підвищення потужності як правило веде за собою збільшення максимальної швидкості та динаміки в певних діапазонах. Бажання людини пересуватися

швидше існувало ще з часів полювання предків на мамонтів. Тому обрана тема, а саме адаптивний алгоритм керування (підвищення ефективності) роботою двигуна автомобіля, буде в усі часи актуальною для людей.

Мета магістерської дисертації – підвищення ефективності (збільшення потужності та зменшення витрат палива автомобілем) за рахунок створення нових програмно-апаратних засобів для реалізації адаптивного алгоритму керування двигуном автомобіля і, робота яких, буде заснована на аналізі характеристик навколишнього середовища та внутрішніх параметрів двигуна, трансмісії й інших вузлів автомобіля.

1. ОСОБЛИВОСТІ ФУНКЦІОНУВАННЯ АПАРАТНОГО ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ БЛОКІВ КЕРУВАННЯ РОБОТОЮ ДВИГУНА АВТОМОБІЛЯ

1.1 ОСНОВНІ ПЕРЕДУМОВИ ДЛЯ ЗБІЛЬШЕННЯ ПОТУЖНОСТІ ДВИГУНА АВТОМОБІЛЯ

Перед розробкою програмного забезпечення для керування двигуном автомобіля варто розуміти, що весь час в ідеальних умовах автомобіль не перебуватиме, і слід передбачити всі фактори, що можуть вплинути на якість та стабільність роботи автомобіля, а саме:

- температура повітря, що входить в автомобіль. Оскільки в сучасному світі лівова частка двигунів оснащені турбокомпресором чи компресором на ремінному або електричному приводі, то необхідно, щоб температура повітря, що двигун втягує в себе, була не високою, але і не низькою для найкращих результатів;
- вологість повітря. Двигуни з турбонаддувом досить примхливі до вологості повітря. Коли повітря вологе, у двигуна з'являється можливість брати додаткове повітря з атомів кисню, що містяться у молекулах води. Але не можна, щоб повітря було надто вологе, краще, щоб повітря було сухе, але з високою щільністю;
- якість пального. Всі двигуни ведуть себе краще на високо-октановому паливі, але не завжди є можливість заправити автомобіль паливом гарної якості. Навіть відомі АЗС, що продають паливо з високим октановим числом, не завжди можуть надати одне і те ж саме паливо з однаковою якістю;
- щільність повітря на вулиці. Чим вище щільність повітря – тим більша кількість повітря, що входить в двигун автомобіля. Але якщо кількість повітря буде надто висока, то повітряно-паливна суміш буде збідніла паливом, що може призвести до оплавлення поршневої групи.
- атмосферний тиск. Чим вище атмосферний тиск – тим вища щільність повітря, і навпаки.

Із-за того, що ці показники не завжди можуть бути такими, на які розраховує інженер, що розробляє ПЗ, виникає необхідність розробляти різне програмне

забезпечення під ті умови, в яких перебувають автомобілі, або робити індивідуальну прошивку під кожний автомобіль, що є надто трудовитратним, і тому якісна програма буде коштувати надто дорого. Питання адаптивного алгоритму та якісної самодіагностики автомобіля є актуальним для найкращого поєднання програмного забезпечення, фізичних процесів роботи двигуна та умов навколишнього середовища.

Насамперед слід розуміти розумні межі того чи іншого допрацювання, оскільки збільшити тиск турбіни та підняти тривалість ін'єкції палива не завжди достатньо, необхідно ще слідкувати за такими показниками, як кількість повітря, кут випередження запалювання, кути відхилення запалювання (детонація), температура випускних газів, температура впускного повітря, кути обертів розподільчих валів, кількість обертів колінчатого валу за хвилину, кількість обертів розподільчого валу, короткотривала та довготривала корекція повітряно-паливної суміші, значення числа лямбда-зонду та великою кількістю інших показників.

Не всі двигуни можна допрацювати за одним і тим самим принципом, деякі двигуни можна доробляти шляхом підняття наддуву та інших супутніх параметрів, деякі двигуни внутрішнього запалення слід робити потужніше шляхом підняття часу тривалості ін'єкції пального в камеру згорання.

Наприклад, у двигунах 2.0 tdi 177 кінських сил (к.с.) сімейства EA189 лише перепрограмуванням електронного блоку управління потужність можна підняти до 240 к.с., але крутний момент зросте всього на 20 Нм. Це обумовлено тим, що максимальний тиск турбіни піднімається всього на 1 psi, що дорівнює 0,07 бар, а тривалість ін'єкції зростає на 30%. Саме тому є така вагома різниця в кількості кінських сил та крутного моменту, що додається.

Тому ідеальний алгоритм, що підійде під всі автомобілі, створити неможливо, в будь-якому випадку потрібні будуть певні правки під конкретний автомобіль, але якщо занести в базу даних ідентифікатори ЕБУ того чи іншого двигуна і врахувати всі їхні особливості, то такий алгоритм працюватиме.

1.2 ВИБІР ДОСЛІДНОЇ МОДЕЛІ ТА ОБҐРУНТУВАННЯ ВИБОРУ

Для тестування алгоритма, що створюється, необхідний автомобіль з достатньою кількістю електроніки в моторі і з досить об'ємною інформацією про електронний блок управління.

Під необхідні потреби було доступно 3 автомобілі:

- Volkswagen Golf 1.9 TDI 105 к.с. з ЕБУ BOSCH EDC16U3х/CP;
- Audi Q5 2.0 TDI 177 к.с. з ЕБУ BOSCH EDC17CP20/CP14;
- Volkswagen Caddy 2.0 TDI 136 к.с. з ЕБУ BOSCH EDC16U3х/CP;

Оскільки двигун 2.0 TDI 136 к.с., працюючий на паливній системі насос-форсунка, є досить ненадійним та погано піддається чіп-тюнінгу, то було прийняте рішення не проводити випробування на автомобілі з цим двигуном.

Автомобіль Audi Q5, оснащений двигуном 2.0 TDI 177 к.с., не підходить під дослідження, оскільки якісно працювати з такими параметрами, як наддув і кількість повітря не можна. Можливість якісно працювати з таблицями управління палива є дуже вагомою перевагою системи Common Rail. Але водночас працювати лише з паливними картами для перших версій адаптивного алгоритму досить проблематично.

Залишається працювати з двигуном 1.9 TDI 105 к.с. та з електронним блоком керування BOSCH EDC16U3х/CP. Цей двигун простий з точки зору чіп-тюнінгу, приріст може скласти понад 40 к.с. та понад 140 Нм крутного моменту лише оптимізацією програмного забезпечення; можна з легкістю піднімати наддув і тримати максимальний наддув хоч у всьому робочому діапазоні двигуна; підготовка пального в форсунках відбувається електронно, а час ін'єкції завжди постійний, оскільки двигун має паливну систему насос-форсункаі, і форсунка відкривається лише від натискання кулачка розподільного валу на рокер; міцність шатунно-поршнєвої групи, що позитивно відображається на надійності ресурсу і зменшує ризики зламати двигун в процесі тестувань; має велику кількість сенсорів для точної діагностики.

В першу чергу необхідно вказати, що двигун 1.9 TDI є одним із найкращих дизельних двигунів. Він витривалий та надійний, як правило, такі двигуни служать від 400 тисяч до 1000000 км. Але є і недоліки. Для випробувань взятий автомобіль з модифікацією двигуна BXE, що була розроблена в останні роки випуску цього двигуна, яка досить дорога в обслуговуванні, запчастини саме на цей двигун знайти складно, але при своєчасному обслуговуванні всіх механізмів буде служити довгий час. Також слід відмітити, що дана модифікація двигуна викликає багато проблем при агресивній експлуатації, що є певною проблемою, оскільки умови, в яких буде працювати автомобіль після перепрограмування, будуть дещо агресивніші.

Також слід відмітити негативні сторони використовуваної моделі, а саме: застарілий принцип керування електро-магнітними клапанами (у випадку виходу з ладу одного клапану, з ладу виходить вся система і це ускладнює діагностику); застаріла електроніка часто дає збій, це робить випробування програмного забезпечення складніше, оскільки чітко виявити недолік стає досить складно; відсутність датчиків у випускній системі, оскільки температура випускних газів є досить необхідним показником; механічне зношування клапана рециркуляції вихлопних газів.

Інформацію з блоку управління BOSCH EDC16U3x/CP можна повністю отримати зчитуванням через діагностичний роз'єм OBD II, але для отримання всієї пам'яті з Flash та EEPROM необхідно демонтувати блок управління та виконувати фізичне підключення до плати електронного блоку керування.

В цілому обрана модель автомобіля є вдалим вибором для випробування адаптивного алгоритму, оскільки є можливість проводити досконально діагностику, мати великий потенціал для підняття потужності шляхом перепрограмування ЕБУ та запас ресурсу. Налаштувавши адаптивний алгоритм не правильно ризик зламати двигун значно менше [1].

1.3 ЦІЛІ ТА РИЗИКИ ПІДНЯТТЯ ПОТУЖНОСТІ

Головною ціллю підняття потужності двигуна автомобіля є покращення динамічних характеристик автомобіля. Як правило на бензинових автомобілях після коректного перепрограмування блока управління зменшується витрата пального, а на сучасних автомобілях, де термостат вже електронний, можна зробити температуру масла двигуна меншою та цим збільшити надійність.

Чіп-тюнінг – це перепрограмування електронного блоку управління двигуна (ЕБУ), з метою поліпшення потужності і динаміки автомобіля без втрати ресурсу, оптимізувавши ККД самого двигуна.

Але збільшити вдвічі потужність лише програмним забезпеченням не можна. Надбавка в 20% можлива тільки для турбодвигунів. Для атмосферного двигуна збільшення можливе в межах 10%. Досягається воно за рахунок більш точного налаштування. Інших способів немає, атмосферний двигун не може в пасивному режимі отримати більше повітря, ніж закладено його виробником.

Турбодвигуни мають більший потенціал для чіп-тюнінгу. Як правило, приріст становить від 15 до 20%. Іноді можна додати навіть 50%, але таке можливо тільки для певних типів двигунів, як 2.0 TSI/TFSI gen3 сімейства EA888.

Також слід приділити увагу двигунам, які вже із заводу виходять з високим ступенем форсування. Наприклад, двигун на Nissan Juke: 1,6 з турбіною, 200 кВт. Для такого об'єму двигуна – це вже майже максимальний показник, і очікувати приросту потужності більше 8-12 кВт в таких випадках не варто.

В ситуації з турбодизелем можна очікувати таку ж надбавку в 20-30%, як і з бензиновим двигуном.

Найчастіше, зустрічається думка, що чим більший робочий об'єм двигуна, тим більший приріст можна отримати після чіп-тюнінгу. Тобто двигун V-вигляду об'ємом в 3 літри може отримати приріст після перепрограмування електронного блоку управління більше ніж звичайний 2-х літровий 4-х циліндровий двигун.

Така думка є помилковою, оскільки максимально можливий приріст залежить від конструкції двигуна, кількості механізмів що можна контролювати

електронікою та закладеному ресурсу заводом виробником. Однаково невеликий приріст потужності можна отримати і від Infiniti (3,7 літрів – не більше 7 кВт), і від Hyundai Solaris (1,6 літрів – 4-6 кВт). Отже, з двигуна меншого об'єму можна отримати більшу надбавку потужності шляхом перепрограмування електронного модулю керування ніж з двигуна більшого об'єму, оскільки важливу роль грає конструктив та механізми, що керуються електронікою.

Також є ризик пошкодити автоматичну або механічну коробку переключення передач в процесі підняття потужності двигуна шляхом чіп-тюнінгу. Це обумовлено значним підвищенням крутного моменту двигуна. Розробники вирішують таку проблему шляхом обмеження крутного моменту програмним забезпеченням.

З підняттям крутного моменту двигуна зростає крутний момент і навантаження на коробку перемикачів швидкостей. Але нема жорсткої залежності, оскільки один і той самий двигун може бути агрегований автоматичною та механічною коробкою переключення передач. У випадку програмного форсування атмосферного двигуна ризик пошкодити коробку переключення передач значно менше, оскільки приріст потужності в 10% зазвичай є мінімально закладеним ресурсом і після підняття потужності шляхом перепрограмування пошкодження трансмісії не відбудеться.

У випадку з дизельними двигунами агрегованими турбокомпресорами крутний момент зростає на 25-70% [2].

Наприклад, із заводу двигун видає під максимальним навантаженням 250 Ньютон на метр крутного моменту. Після перепрограмування крутний момент може зрости до значення 450 Нм. При розробці програмного продукту інженер повинен враховувати максимально допустиму потужність на якій може працювати двигун та коробка передач без шкоди для вузлів агрегатів.

Сучасні трансмісії, як правило, розраховані на роботу з потужністю що на 150% більше за заводську. Також завод виробник програмно обмежує максимальний крутний момент з ціллю збереження механізму. В такому випадку при перевищенні двигуном обмежувача крутного моменту коробка передач

викличе появу помилку в шині CAN і переведе автомобіль в аварійний режим. Щоб такого не відбувалося інженери приховують справжній крутний момент від електронного блоку управління автоматичної або роботизованої коробки переключення передач.

Існує твердження, що тюнінгу без зниження ресурсу не буває. Все залежить від того, який стоїть двигун, який у нього із заводу закладений запас, і який взагалі передбачається чіп-тюнінг. Так, зростання моменту здатне знизити ресурс агрегатів, але головний фактор такий, як водій буде розпоряджатися отриманою надбавкою потужності. Якщо він їздить в режимі “газ в підлогу – холостий хід”, то надбавка в 20% на ті ж 20% скоротить ресурс двигуна [2]. А ось якщо зберегти спокійну манеру, і отриману надбавку використовувати тільки при необхідності (наприклад, при обгонах), то вік двигуна не стане коротшим.

Однак, слід зазначити, що люди, які збільшують потужність свого автомобіля, роблять це не для спокійної їзди. З іншої сторони, випадків поломок на автомобілях з найменшим рівнем прошивки значно більше ніж з максимальним, оскільки власник автомобіля з високою потужністю приділяє більшу увагу своєму авто та надає краще обслуговування, ніж власник автомобіля з більш слабкими характеристиками після допрацювання. Тому увага власника до автомобіля та рівень його обслуговування грає ключову роль в надійності автомобіля як до, так і після підвищення потужності.

Також слід враховувати пробіг та мотогодини роботи двигуна. Автомобіль може мати малий пробіг, але матиме багато мотогодин роботи, це означає, що автомобіль майже не буває під навантаженням, а певні агрегати не працювали в необхідних для них режимах.

При розумному чіп-тюнінгу ризиків менше, ніж користі. Є випадки, коли компанія максимально піднімає наддув турбіни автомобіля, при цьому не змінюючи всі інші параметри, а датчики, що впливають на безпеку, відключають, і отримують значно нижчі результати, ніж прошивка з меншим тиском повітря, з включеними всіма датчиками безпеки. Це обумовлено кардинально різними підходами до підняття потужності автомобіля.

Головним ризиком при чіп-тюнінгу є людський фактор. Є люди, які розробляють самі програмне забезпечення і вміють це робити якісно, є просто дилери програмного забезпечення, які не мають доступу до вихідного коду і не можуть їх правити, Якщо ціль перепрограмування – отримання надійного та протестованого рішення, то слід звертатися до іменитих компаній, що гарно зарекомендували себе на ринку чіп-тюнінгу. Якщо необхідне особисте налаштування, то краще всього це зроблять інженери, що займаються виключно індивідуальним калібруванням механізмів двигуна, що керуються електронікою. Як правило, така необхідність є лише для автомобілів, що беруть участь у змаганнях.

Як правило всі ці поломки відбуваються на не протестованих програмах і тому якісними прошивками слід вважати ті, що були протестовані не на клієнтських автомобілях, а на автомобілях компанії перед тим як пропонувати клієнтам спробувати їх програмне забезпечення.

1.4 ОСОБЛИВОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ БЛОКІВ КЕРУВАННЯ ДВИГУНОМ АВТОМОБІЛЯ

Розглянемо підходи 4-х різних розробників програмного забезпечення, а саме APR, Revo Technik, Superchips, Sedox performance. Ці компанії є одними із найпопулярніших компаній у світі.

Sedox Performance. Порівняння слід почати саме з цієї компанії. Існує вебсайт, на якому розробники можуть розміщати своє програмне забезпечення, за кожне таке розміщення розробник заробляє \$30, а користувач за кожне скачування платить \$30. Це програмне забезпечення тестується на клієнтських автомобілях і ніколи не виправляється, адже не відомо, хто розробник, і за якість такого програмного продукту ніхто відповідальності не несе.

APR. Відомий розробник якісного програмного забезпечення з дуже розвинутою дилерською мережею. Працює виключно з автомобілями групи VAG.

Не зважаючи на масштабність та можливості компанії, їх програмний продукт не завжди є якісним, хоча показники, що показують автомобілі на прошивці цього виробника, є дуже високими. Оскільки компанія створена на ринку США, то їх продукт максимально підходить для автомобілів для ринку США. Досягає високих результатів шляхом високого тиску повітря, що досить негативно впливає на двигун та на стабільність результатів. Компанія також розробляє своє апаратне забезпечення для високих рівнів допрацювань.

Revo Technik. В свій час компанія не розробляла програмне забезпечення, а була дилером APR. Оскільки APR робить прошивки для ринку Європи гірше, ніж для ринку США, було прийняте рішення відділитися та заснувати свою компанію. За 15 років існування має дуже велику мережу дилерів по всьому світу, 5 із них в Україні. Компанія знаходиться у Великій Британії, тому її продукти краще всього підходить під автомобілі європейського ринку. Розробляє прошивки для автомобілів групи VAG та Ford. Має дуже стабільні показники, а систем захисту двигуна більше, ніж в заводських програмах. Навіть на несправних автомобілях демонструє себе досить стабільно та якісно, чим і зарекомендувала себе з позитивної сторони. Потужність досягається великими кутами запалювання та бідною паливно-повітряною сумішшю, що є досить ризиковано, але при цьому найнадійніше для двигуна автомобіля. Програмне забезпечення розробляється від 9 місяців і до 2-х років, оскільки вноситься дуже багато корективів і налаштовується під умови кожної країни, в яких є дилер; цих регіонів більше 72. З однієї сторони розробник змушує чекати клієнта до 2-х років для отримання програмного забезпечення, а з іншої сторони дає продукт, що протестовано та має високу якість.

Superchips. Одна із перших компаній в світі, що займається розробкою програмного забезпечення для двигунів автомобілів. Існує з 1976 року та знаходиться у Великій Британії. Спеціалізується на 52-х марках автомобілів. Також має розвинуту мережу дилерів по всьому світу, з них 2 в Україні. Компанія відома своєю історією в мотоспорті, розробляє програмне забезпечення не лише для автомобілів, а і до мотоциклів. Має схожий підхід до розробки програми, як і Revo Technik, але суміш робить багатішою на паливо, тим самим робить програмне

забезпечення слабкішим та еластичнішим для щоденного використання. Також є абсолютним лідером по розробці програмного забезпечення для дизельних автомобілів. Ефективність та якість прошивок для дизелів перевершує всіх конкурентів. Програмне забезпечення має однакову якість як для автомобілів з ринку США, так і для європейського ринку, оскільки розробляє програму індивідуально під кожний автомобіль та може вносити зміни в продукт, якщо є необхідність або побажання клієнта.

1.5 НЕДОЛІКИ ІСНУЮЧИХ РІШЕНЬ

Головними недоліками існуючих рішень є вимога програмного забезпечення до апаратного працювати завжди в тих умовах, які були прописані розробником. Нездатність виконати вимоги програми може бути пов'язана не лише з несправністю агрегатів, але і з умовами навколишнього середовища. У випадку невиконання вимог програми налаштування переходять в аварійний стан, тим самим зменшуючи потужність автомобіля.

Такий підхід дієвий та дуже безпечний для двигуна автомобіля, але є надто радикальним. Водію необхідно чекати протягом 40-80 км на бензинових двигунах і 120-150 км на дизельних двигунах, щоб програма дозволила підняття потужності [2].

Не всі програмні засоби мають можливість проводити діагностику роботи двигуна, оскільки деякі розробники відключають датчики, що необхідні для коректного утворення суміші і слідкування за детонацією всередині циліндра. При діагностиці роботи двигуна втрачається можливість зробити правильний висновок, оскільки незрозуміло, що відбувається всередині двигуна.

Деякі розробники, наприклад, як AGP Motorsport, піднімає потужність двигуна автомобіля лише налаштовуючи електромагнітний клапан турбіни, а всі інші параметри автомобіль сам адаптує під роботу турбіни. Таке налаштування є максимально небезпечне для автомобіля, оскільки навіть у випадку поломки турбіни актуатор буде змушувати працювати турбіну саме так, як це прописав

розробник. Якщо буде несправний якийсь інший механізм, то програмі буде так само байдуже, оскільки вона працює лише з одним параметром двигуна. Також ця компанія відключає датчики детонації та всі інші системи безпеки, щоб досягти максимально кращого результату динаміки, але це робиться лише ціною двигуна автомобіля.

Певні принципи адаптації можна відслідкувати у програмних продуктах компанії Revo. Оскільки інженери витрачають на розробку прошивки інколи декілька років та закладають багато систем захисту в свої програми, то не дивно, що їхні продукти мають адаптивний характер.

Відразу після інсталяції програми в електронний блок керування потужність піднімається на 70% від максимально можливого приросту. Вся потужність піднімається на 100% лише після 150 км, якщо нема ніяких негативних факторів.

У випадку детонації чи недостатнього тиску турбіни програма активує аварійну карту значно раніше, ніж це буде зроблено на заводській програмі. Якщо умови роботи агрегатів повернуться до робочого стану, то програмне забезпечення знову поступово буде піднімати потужність автомобіля.

Саме тому інженери вимагають проводити діагностику відразу після оновлення програмного забезпечення блоку управління двигуна, щоб переконатися в тому, що нема ніяких негативних проявів роботи.

Оскільки програмісти розробляють програмне забезпечення лише на одному блоці управління, то для інших версій модулів програму просто корегують; такий підхід не завжди підходить і тому виникають помилки. В таких випадках потрібне індивідуальне налаштування певної версії електронного модуля керування.

Виробник програмного забезпечення Superchips також має деякі алгоритми, що роблять їхні продукти адаптивними, але їх значно менше, оскільки інженер на розробку витрачає всього декілька тижнів.

Так само як і у Revo, програма відразу дає лише 70% свого потенціалу, а вся інша потужність піднімається лише з часом. Superchips відразу переводить управління двигуном в аварійний режим у разі виникнення якоїсь важливої помилки, при появі детонації в циліндрах програма просто зменшить кут

запалювання. Повноцінне функціонування повернеться вже після перезапуску двигуна, програма повернеться до стандартної карти і продовжить роботу до появи наступних помилок в ключових агрегатах двигуна.

Методи адаптації програмного забезпечення роботи двигуна автомобіля з ціллю підняття потужності та збереження ресурса автомобіля вже існують, але вони недосконалі і працюють лише за заданим розробником алгоритмом.

Недоліками такого підходу є обов'язкова діагностика спеціаліста, щоб з'ясувати, які поломки є в двигуна. Програма не здатна налаштовувати сама себе і завжди працюватиме лише за заданими параметрами. Єдине, що може зробити програма – перейти в захисний режим. Це відбувається, лише якщо якийсь механізм не може виконувати ті параметри, що були задані розробником і лише за умови, що ці захисні режими були створені на етапі розробки.

Існуючі рішення вимагають роботи двигуна саме в тих параметрах, що задають інженери. З одного боку це добре, оскільки в програми є рамки і зрозуміло, як потрібно працювати. А з іншого боку, це може шкодити ресурсу двигуна та коробки передач, оскільки не всі розробники розумно створюють свій програмний продукт, що може призвести до негативних наслідків.

Метою магістерської дисертації є підвищення потужності автомобіля за рахунок створення алгоритма, що автоматично підвищує необхідні параметри і не шкодить безпеці двигуна, адже у всіх водіїв в першу чергу стоїть питання довготривалої роботи без шкоди для двигуна та коробки перемикачів передач.

Висновки до розділу 1

В розділі 1 були розглянуті всі існуючі програмні рішення, що існують на сьогоднішній день, програмного забезпечення що підвищує потужність двигуна шляхом оптимізації управління механізмів двигуна, що керуються електронікою.

Проведено порівняння реалізованих програмних засобів що є лідерами в цій галузі, що розглядається. В результаті проведеного огляду показано, що для якісного програмного засобу необхідно мати інформацію не лише про можливості механізмів для яких буде змінено керування, а й інформацію про параметри, що впливають на ресурс, який був закладений заводом виробником.

Основаючись на існуючих програмних рішеннях було прийняте рішення розробити програмний засіб підвищення ефективності шляхом створення нових програмно апаратних засобів.

1. СПОСОБИ ПІДВИЩЕННЯ ЕФЕКТИВНОСТІ РОБОТИ ДВИГУНА

2.1 СПОСОБИ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ДВИГУНА АВТОМОБІЛЯ

Є багато різних способів підвищення потужності автомобіля. Деякі розробники переписують 90% карти роботи двигуна, а хтось просто робить відкриття дроселя раніше, тим самим створюючи враження потужнішої машини.

Розробники програмного забезпечення для двигуна автомобіля, як правило, використовують один з наступних підходів: піднімають потужність коефіцієнтами; піднімають потужність шляхом описування моделі; працюють лише з одним параметром; індивідуальне налаштування на основі показників лямбда-зонду.

В алгоритмі, що розробляється в даній магістерській дисертації, підвищення потужності буде відбуватися шляхом підняття коефіцієнтів в повітряних та паливних картах роботи двигуна. Кожна таблиця матиме свій коефіцієнт, що буде залежати від коефіцієнтів в інших таблицях та від повітряно-паливної суміші.

Така реалізація значною мірою ускладнить діагностику коректності роботи програмного продукту, оскільки кожного разу програма буде працювати по-різному. Але це максимально проявить некоректну роботу певного механізму, оскільки у випадку незмоги підвищити навантаження помилка буде відразу відображена в списку помилок.

2.2 Підвищення потужності за допомогою таблиці коефіцієнтів

Даний підхід може бути застосований в реалізації адаптивного алгоритма, оскільки являється максимально зручним механізмом поступово підвищувати та зменшувати потужність двигуна автомобіля у випадку необхідності.

Відкриваючи карту паливну або повітряну, спеціаліст просто множить всі заводські параметри на певний коефіцієнт і пророблює таку дію для всіх карт. Тим

самим він підвищує потужність автомобіля, але також зростає навантаження на всі агрегати.

Наприклад: із заводу розробник задає значення тиску повітря після турбокомпресора 0,8 бар, при множенні цього значення на коефіцієнт 1,5 тиск стає 1,2 бар. Кількість впорскування пального в циліндр 56 мг на цикл в заводському програмному забезпеченні. Після множення цього значення на коефіцієнт 1,5 об'єм пального, що впорскується, стає 98 мг на цикл.

З однієї точки зору, все повинно працювати як слід, оскільки суміш залишилась така ж сама, як визначено заводом-виробником, збільшення робочого тиску турбіни та кількості пального, що було подано в циліндр, в півтора рази, не зможе нашкодити такому двигуну, як 2.0 TSI gen2 сімейства EA888, оскільки такий двигун розрахований витримувати потужність в 3 рази більшу, ніж було реалізовано із заводу. Так само це не працює на дизельних двигунах, де після перепрограмування електронного блоку управління двигуна крутний момент збільшується в півтора рази в порівнянні із заводськими показниками.

З іншої точки зору, навантаження на механізми в півтора рази не дасть приросту потужності в півтора рази. Це обумовлюється тим, що окрім параметрів тиску повітря перед дросельною заслонкою та кількості пального, що впорскується в циліндр, є ще такі параметри, як кут запалювання, цей параметр інколи краще робити меншим, ніж із заводу, за умови налаштування роботи автомобіля на пальне низької якості, кількість повітря, що поступає в двигун, тривалість ін'єкції пального, тиск палива в паливній рейці.

Це створює дискомфорт при користуванні автомобілем, оскільки втрачається вся еластичність автомобіля у всіх діапазонах роботи двигуна. Так як автомобіль починає працювати весь час під більшим навантаженням, всі механізми підвищують навантаженість роботи також в півтора рази, це значною мірою шкодить ресурсу автомобіля та підвищує зношування всіх вузлів силового агрегату автомобіля. Витрата пального також зростає в півтора рази.

Тому такий підхід є непрофесійним, адже лише шкодить ресурсу двигуна та створює багато проблем для власника автомобіля. Потужність підвищується і такий

підхід дає дієвий результат після перепрограмування блоку управління таким чином.

Підвищувати потужність шляхом коефіцієнтів можливо, але це потрібно робити іншим чином. Потрібно виставляти різні коефіцієнти для різних таблиць, в яких розробник вказує задані параметри роботи двигуна. Інколи краще зробити тиск повітря перед дросельною заслонкою менше, але кут запалювання повітряно-паливної суміші в циліндрах більшим, це буде значною мірою важче в реалізації, але в рази якісніше з точки зору підняття потужності. Це значно важчий та довший процес, але якісно отримати вищу потужність, не втрачаючи в якості програмного продукту та ресурсу шляхом множення заводського значення на коефіцієнт, можливо лише таким чином.

2.3 Підвищення потужності шляхом описування моделі

Цей спосіб є максимально професіональним та точним. Спеціаліст в кожній карті описує модель, яку хоче досягти, а модулі керування агрегатів підвищують свою продуктивність з ціллю отримання тих результатів, що були описані розробником.

Наприклад, інженер описує тиск повітря у впускному колекторі в момент часу та описує, за яких умов це має відбуватися. Такі величини називаються заданими параметрами. Значення, що виконує двигун на вимогу програмного забезпечення, називаються фактичними параметрами. Якщо у автомобіля є датчик масової витрати повітря, то відповідно до кількості повітря, що перебуває у впускному колекторі, готується необхідна кількість пального, щоб співвідношення повітря до пального відповідала моделі стехіометричного числа, що було прописано інженером заздалегідь. Якщо сенсор кількості повітря в автомобілі відсутній, то відповідно до математичної моделі електронний блок управління двигуна вираховує, яка кількість повітря потрапляє до двигуна. Якщо фактичне значення не відповідає заданому, то двигун сам буде піднімати чи опускати

навантаження на той чи інший механізм, що впливає на той чи інший параметр роботи.

Таким чином, інженер отримує можливість тонко налаштовувати роботу програмного забезпечення для конкретного двигуна. Маючи точно описані таблиці керування повітрям та повітряно-паливної суміші, можна вже налаштовувати інші параметри, що також впливають на потужність або ресурс двигуна.

При розробці такого програмного продукту інженеру слід знати, який механізм на що здатний і відповідно до цього будувати модель, що не шкодитиме ресурсу двигуна. Якщо той чи інший агрегат не може виконати те навантаження, що від нього вимагається, то він несправний, а не погано створене програмне забезпечення. Це убезпечує двигун від завчасного зношення та пошкодження.

Також це дає унікальну можливість створювати аварійні карти, що будуть виконуватися у випадку тих чи інших неполадок в автомобілі. Одним із таких режимів є обмеження потужності двигуна, якщо температура температурного масла нижче за 70 градусів. Така робота може зайняти далеко не один місяць роботи і буде коштувати великих грошей, тому індивідуально розробляти таке програмне забезпечення під кожний автомобіль не є доцільним з точки зору часу та фінансів.

Поширення такого програмного забезпечення відбувається шляхом адаптації існуючого програмного продукту під всі інші блоки управління такого ж двигуна. Цей підхід дієвий, але не для всіх електронних модулів управління. Для певних блоків необхідно більш індивідуальне налаштування не зважаючи на те, що механічно двигун буде таким самим.

Як правило, адаптивність такого способу обмежується лише активацією аварійних карт.

Такий спосіб створення програмного забезпечення має розроблятися суто під конкретний вид двигуна і не може бути використаним для розробки адаптивного алгоритму, незважаючи на велику кількість позитивних сторін. Проблематичність впровадження такого способу під цілі магістерської дисертації полягає в тому, що програма в автоматичному режимі не зможе швидко описувати всі карти під ті

умови, в яких знаходиться, та перезаписувати багато разів. Це буде використовувати багато обчислювальних потужностей процесора електронного блоку керування.

2.4 Підвищення потужності за допомогою лише з одного параметру

Деякі компанії працюють лише з одним параметром двигуна, таким як положення електромагнітного клапану турбіни або просто із тиском турбіни, а всі інші показники підвищуються, щоб відповідати заданому параметру.

Наприклад, на сучасних двигунах, що оснащені турбокомпресором, керування турбіною відбувається шляхом подання певної напруги на електромагнітний клапан турбіни. Це дає змогу точно керувати тиском повітря. Задані параметри тиску турбіни будуть підняті автоматично, оскільки буде змінено заданий параметр закриття регулювального клапану турбіни. Оскільки стехометричне число залишається таким, як було задано на заводі, то блок керування двигуном дає команду збільшити впорскування пального до циліндрів. Кут запалювання також зберігається заводським, оскільки інженер не змінює цей важливий показник.

В результаті виходить чітке управління турбокомпресора, але не контролюються інші фізичні процеси двигуна. Можна досягти високих результатів динаміки автомобіля. У автомобіля буде правильна суміш, передбачена заводом-виробником, та невисоке збільшення навантаження на певні механізми.

Переваги такого підходу: точне управління турбіною автомобіля з ціллю контролювати тиск повітря. Залишаються заводські налаштування всіх інших параметрів, що прибирає людський фактор в налаштуванні. Автомобіль буде проявляти потужність лише під навантаженням, у спокійних режимах поведінки автомобіля не зміниться в порівнянні із заводським програмним забезпеченням. Високі показники динаміки автомобіля.

Недоліки даного способу: інженер не може вплинути на роботу інших вузлів, тому на таких налаштуваннях дуже висока ймовірність поломки двигуна. Якщо

збільшувати кількість обертів турбінного колеса, то через певний час доведеться замінити весь турбокомпресор, оскільки колесо здатне пошкодити корпус турбіни, також є ризик того, що залишки компресорного колеса попадуть до двигуна автомобіля та пошкодять його. Оскільки тиск турбіни можна контролювати не лише відсотком закриття регулювального клапану турбокомпресору, а і кількістю випускних газів із двигуна автомобіля. В таких програмах, як правило, відключені системи захисту, тому що блок керування не міг точно керувати всіма параметрами роботи двигуна по тій причині, що розробник працює лише з одним параметром, ігноруючи інші. Це може допомогти отримати високий результат, але це буде лише обмежену кількість разів. Ці результати не будуть стабільними і з часом ставатимуть все гірше. Це буде проявом зношування турбокомпресору.

Аналізуючи роботу двигунів на таких програмах було помічено, що розробник має змогу керувати роботою двигуна лише у певних межах, і відбуваються різкі переходи між керуванням розробленої програми та заводським керуванням.

Цей спосіб взагалі не має адаптивності, оскільки просто виконує задані інженером параметри.

Такий спосіб може бути застосований для розробки адаптивного алгоритму з метою підняття потужності автомобіля, але не варто використовувати. Це надто небезпечно для двигуна.

Цей підхід є прикладом неправильного налаштування роботи двигуна, тому його не буде використано.

2.5 ІНДИВІДУАЛЬНЕ НАЛАШТУВАННЯ НА ОСНОВІ ПОКАЗНИКІВ ЛЯМБДА-ЗОНДУ

Лямбда-зонд це сенсор, що міряє показники випускних газів та аналізує залишковий вміст кисню або бензину з ціллю визначення, збагачена чи збідніла суміш на бензин. Цей спосіб був створений ще в ті часи, коли розробники не знали, що таке стехіометричне число і звертали увагу лише на кінцевий результат після згорання паливно-повітряної суміші. В перші часи замість заводського лямбда-

зонду вкручували власні сенсори вимірювання випускних газів. Головною проблемою було те, що блок управління двигуна все одно продовжував працювати за показниками, що фіксувалися заводським лямбда-зондом і тому двигун працював некоректно, відповідно, таке налаштування несло лише негативний характер для двигуна. З часом, коли двигуни стали більш технологічними і електроніка почала керувати більшою кількістю процесів, розробники почали слідкувати за показниками виключно штатного лямбда-зонду, таким чином якість розробленого програмного забезпечення значно зросла.

Інженер власноруч налаштовує наддув та впорскування пального, описуючи модель роботи різних параметрів або застосування способу множення на коефіцієнти заданих параметрів роботи. Розробник має слідкувати за стехіометричним числом та реальною корекцією паливно-повітряної суміші. Це дуже гарний підхід з точки зору індивідуального створення програмного забезпечення, адже кожний по-різному експлуатує свій автомобіль, але якщо автомобіль має якусь несправність, то інженер некоректну роботу двигуна зробить нормою і тим самим замаскує цю поломку.

Переваги такого підходу: інженер може налаштувати роботу двигуна автомобіля конкретно під вимоги замовника. Суміш автомобіля буде налаштовано максимально правильно і не нестиме негативних наслідків для двигуна. Якщо автомобіль повністю справний на момент налаштування, то в результаті двигун буде налаштовано повністю коректно і буде звичайним в діагностуванні. Навантаження на механізми буде залежати виключно від того, наскільки інженер буде підвищувати потужність.

Недоліки такого підходу: в першу чергу такий програмний продукт матиме один великий ризик, який буде полягати в тому, що все буде залежати виключно від кваліфікації розробника, що створюватиме таке програмне забезпечення. Якщо на момент налаштування автомобіль буде несправний і розробник не зафіксує це, то налаштуванням такого двигуна буде мати багато проблем, оскільки після усунення несправності робота двигуна буде відбуватися в зовсім інших діапазонах, ніж були прописані інженером. Тому після кожних виправлень поломок доведеться

змінювати налаштування програми керування двигуна. Проводити діагностику двигуна на такому налаштуванні може лише той, хто створював цю програму або повернувши блок управління двигуна до заводських параметрів.

Такі продукти взагалі не адаптивні і можуть працювати лише так, як це передбачив розробник. Така програма може мати адаптивність лише якщо інженер власноруч розробив способи оптимізації.

Спосіб розробки програмного забезпечення шляхом індивідуального налаштування на основі показників лямбда-зонду буде використано лише частково при розробці адаптивного алгоритму. Показники лямбда-зонду є дуже важливим показником і саме цей сенсор може вказати на несправність двигуна у випадку, якщо стехіометричне число буде в нормі.

Висновки до розділу 2

У розділі 2 було розглянуто всі існуючі способи що використовують для покращення характеристик потужності двигуна та проведено загальний аналіз таких підходів.

Розглянуто такі способи, як:

1. Підвищення потужності коефіцієнтами
2. Підвищення потужності шляхом описування моделі
3. Підвищення потужності лише з одним параметром
4. Індивідуальне налаштування на основі показників лямбда-зонду

На основі проведеного аналізу були оцінені всі переваги та ризики того чи іншого підходу.

В ході аналізу було вирішено яким способом буде розроблятися адаптивний алгоритм, а саме шляхом множення заводських параметрів на коефіцієнти, кожний параметр повинен мати свій особливий коефіцієнт та буде мати можливість змінюватися у реальному часі в залежності від стилю керування автомобілем.

2. ПРОГРАМНІ ЗАСОБИ ДЛЯ ВНЕСЕННЯ ЗМІН В ПАРАМЕТРИ КЕРУВАННЯ АВТОМОБІЛЕМ

3.1 ОСОБЛИВОСТІ ЗАСТОСУВАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ В АВТОМОБІЛІ

Програмне забезпечення для автомобілів відносно надійне - рівень відмов становить не більше одного збою на мільйон операцій в рік. Більшість людей навіть не уявляють, наскільки багато автомобільних функцій керуються сьогодні програмно.

Зараз кожен автомобіль має декілька електронних блоків управління, пов'язаних між собою мережею в середині автомобіля. Ці блоки взаємодіють через стандартні шинні архітектури, такі як мережа контролерів (controller area network, CAN), мережа передачі даних мультимедійних систем (media-oriented systems transport, MOST), FlexRay і локальний інтерконект (local interconnect network, LIN). У порівнянні з Ethernet, широко використовуваним для зв'язку ПК, перераховані шини працюють повільніше - в автомобілях обсяг інформації, що пересилається невеликий, але її необхідно обробити за кілька мілісекунд. Збільшення числа електронних блоків управління, що пов'язуються, приводить до необхідності створення більш складних структур мереж в середині машини, які потребують особливої електричної та електронної архітектури. Відмінності між автомобільним програмним забезпеченням та іншими видами ПЗ:

- надійність: автомобільні програмні системи повинні працювати виключно надійно в складній мережі електронного блоку керування протягом всього терміну експлуатації автомобіля;
- функціональна безпека: такі функції, як антиблокувальна гальмівна система і ESC, вимагають безвідмовної роботи, що визначає високі вимоги до процесів розробки програмного забезпечення і до самих програм;
- робота в режимі реального часу: швидка реакція (від мікросекунд до мілісекунд) на зовнішні події вимагає оптимізованих операційних систем і особливої програмної архітектури;

- мінімальне споживання ресурсів: будь-яке доповнення обчислювальних ресурсів або пам'яті збільшує вартість продуктів, що при мільйонних тиражах виливається в чималі гроші;
- надійна архітектура: автомобільне програмне забезпечення повинне витримувати спотворення сигналів і підтримувати електромагнітну сумісність;
- електронно-механічне управління замкнутого циклу.

При цьому треба врахувати, що перезавантаження під час роботи для більшості електронних модулів керування неприпустима.

Якщо в перші роки появи автомобільного ПЗ його міг контролювати один розробник, то тепер це вже неможливо.

У 70-х роках розробники програмного забезпечення для автомобілів почали використовувати асемблер, а С стала основною мовою в 90-х роках. Протягом останнього десятиліття компанія Robert Bosch і інші постачальники автомобільних компонентів стали розробляти програмне забезпечення на базі моделей, використовуючи ASCET (вдосконалений інженерний інструментарій моделювання та управління) і Mathlab / Simulink [3].

Шинні системи, такі як CAN, серйозно ускладнюють програмне забезпечення, оскільки допускають взаємодії між програмами різних електронних блоків управління. В автомобілях класу люкс складна мережа пов'язує зараз до 80 ЕБУ, в сукупності мають до 100 млн рядків коду. Оскільки програмне забезпечення стає все складніше, виникає необхідність удосконалювати методи інжинірингу, відповідно в галузі сьогодні пропонуються паралельні організаційні і технічні процеси для розробки програмного забезпечення. Компанія Bosch давно застосовує розробку на базі процесів інжинірингу та управління, відповідних СММІ рівня 3, а її інженерний підрозділ в Індії вже досягли рівня 5.

Сьогодні на частку електроніки припадає близько 80% функціональних інновацій автомобільної галузі, і програмне забезпечення - це ключ до більшості з них. У міру того як програмне забезпечення стає все більш істотною частиною вартості обладнання, в бізнес-моделях починають враховувати необхідність повторного використання та обміну програмним забезпеченням.

Високошвидкісні шини, такі як Ethernet, все ширше використовуються сьогодні в автомобілебудуванні для підтримки взаємодії між електронними блоками керування і розробки нових функцій, особливо в області безпеки. Інформація з різних джерел аналізується і консолідується для формування повної моделі середовища, дозволяючи розробляти нові функції, які підтримують водія в критичних ситуаціях. Наприклад, якщо увагу водія відволікає пасажир, то програма може визначити, що їде попереду гальмує, і попередити про це водія або ж автономно включити гальмування. Водій ніколи не здогадається про існування такого програмного забезпечення, поки не виникне небезпечна ситуація.

В автомобілебудуванні сьогодні назріла чергова програмна революція - все ширше починають застосовуватися засоби мультимедіа та побутової електроніки. Автомобілі будуть підключатися до Інтернету і до всіх видів мобільних і встановлених будинку пристроїв, причому неухильно зростатиме частка рішень на базі вільного програмного забезпечення [4].

3.2 ВИКОРИСТАННЯ ТЕХНОЛОГІЇ AUTOSAR

При розробці сучасних цифрових програм для автомашин однозначно враховується специфіка необхідного ЕБУ: програмне забезпечення безпосередньо взаємодіє з певним обладнанням. Завдяки постійному зростанню кількості автомобільних електронних блоків управління, вторинне використання ПЗ стає в пріоритеті. Тому в такій ситуації доречно говорити про стандартизацію [5].

У 2003 році постачальниками і виробниками було створено об'єднання «Automotive Open System Architecture» (Autosar). Мета створення організації - виконання загального стандарту і єдиних технологій. Сьогодні це об'єднання охоплює понад 150 організацій, якими спільно розробляється нова будова ECU, базове ПЗ і все необхідне для створення робочого програмного забезпечення.

Такого роду взаємодія передбачає створення вузлів, що не залежать від обладнання. Це дає можливість постачальникам і виробникам обмінюватися

розробками, а також використовувати їх повторно на самих різних електронних модулях керування.

Будова Autosar складається з декількох абстрактних рівнів, в яких програмне забезпечення відділяється від апаратного забезпечення. На самому верху знаходиться прикладне ПЗ, яке реалізує всю прикладну діяльність. Нижче слід базове, номінальне програмне забезпечення. Воно гарантує потрібну абстракцію від апаратного забезпечення точно так само, як це відбувається, наприклад, в персональному комп'ютері. Autosar Runtime Environment (середовище виконання в реальному часі) здійснює зв'язки всередині електронного блоку керування.

Технологія Autosar містить всі необхідні формати обміну і шаблони, які використовуються як для генерації і конфігурації інфраструктури, так і для її опису.

Найрізноманітніша інформація якісно аналізується з метою створення об'єктивної моделі навколишнього середовища, що дозволяє формувати нові опції, які підтримують водія в екстремальних випадках [6].

AUTOSAR - переслідує мету створити та встановити відкриту та стандартизовану архітектуру програмного забезпечення для автомобільних електронних блоків управління (ЕБУ). Цілі включають масштабованість до різних варіантів транспортних засобів та платформ, можливість передачі програмного забезпечення, врахування вимог щодо доступності та безпеки, співпрацю між різними партнерами, стале використання природних ресурсів та ремонтпридатність протягом усього життєвого циклу продукту.

Стандарт AUTOSAR визначає набір підтримуваних розподілених, керованих функціями методологій розробки програмного забезпечення для автомобільної електроніки та програмного забезпечення на електронних блоках управління, стандартизованих рішень із додатками в різних автомобілях та платформах для поліпшення повторного використання програмного забезпечення та зменшення витрат на розробку [7].

До теперішнього часу дослідження та розробка систем AUTOSAR все ще перебуває на стадії попереднього техніко-економічного обґрунтування, ще не вступили у фактичну фазу розвитку.

Крім того, Сполучені Штати співпрацюють для надання Freescale розробки програмного забезпечення для управління автомобілем, сумісного з Elektrobit AUTOSAR, впровадження програми. Компанії MathWorks та Vector Informatik співпрацюють для досягнення взаємної сумісності інструментів додатків AUTOSAR.

Automotive Software Platforms		
Segments	Key Information	Software Platform Info & Examples
Control program & Operating system (OS)	<ul style="list-style-type: none"> ECU control software from auto industry OS is mostly from high-tech industry OS for driving control needs certification 	<ul style="list-style-type: none"> AUTOSAR platform; MCUs to ECUs Linux: AGL, Android, GENIVI, others QNX, Green Hills, Wind River
Hypervisor OS	<ul style="list-style-type: none"> OS to combine infotainment & safety systems Ex: Infotainment & backup display 	<ul style="list-style-type: none"> Green Hills, QNX, Wind River, others Strong growth from domain ECUs
Instrument cluster system	<ul style="list-style-type: none"> Dominated by Tier 1 suppliers Becoming part of display infotainment 	<ul style="list-style-type: none"> Major OEMs have proprietary versions Often using OS such as QNX or Green Hills
Telematics system & Telematics services	<ul style="list-style-type: none"> In-car software client & TSP SaaS platform Software platforms from TSP, Tier 1 or OEM OS from high-tech industry 	<ul style="list-style-type: none"> Most software platforms are proprietary OnStar, SiriusXM, Verizon, WirelessCar QNX is leading telematics OS
Infotainment systems (Also called Head-Units)	<ul style="list-style-type: none"> Most complex auto software system currently Display, music, navigation, apps & HMI Adding smartphone integration & VPA Requires OS such as QNX or Linux version 	<ul style="list-style-type: none"> Display-based systems is largest category Dominated by Tier 1 suppliers such as: Harman, Panasonic, Bosch, Continental, Denso, Marelli, LG Electronics, Visteon
Smartphone apps integration	<ul style="list-style-type: none"> Growing source for infotainment & content Apple and Google are becoming dominant 	<ul style="list-style-type: none"> Apple CarPlay, Android Auto MirrorLink, Smart Device Link, Baidu Carlife
Virtual Private Assistant (VPA)	<ul style="list-style-type: none"> Function control via speech user interface Amazon, Apple & Google are dominant 	<ul style="list-style-type: none"> Amazon Alexa, Apple Siri Google Assistant and others
AUTOSAR=Automotive Open System ARchitecture; AGL=Automotive Grade Linux; TSP=Telematics Service Provider		

Рисунок 1 – Автомобільні програмні платформи

Платформа - це технологія, що дозволяє створювати схожі між собою продукти, що використовують результати минулих розробок. В рамках стратегії реалізації платформ багато продуктів або сімейства продуктів можуть бути розроблені з набагато меншими витратами, ніж при розробці кожного продукту окремо. В автомобільній промисловості широко використовуються поширені шасі з різними двигунами, салонами і форм-факторами. Платформи BEV в даний час стають важливою тенденцією в автомобільній промисловості. Крім того, цей підхід дуже поширений в комп'ютерній індустрії - часто можна зустріти платформи різних рівнів (починаючи з процесорних і апаратних, і закінчуючи програмними платформами і платформами додатків) [8].

Використання програмних платформ стало ключовою стратегією у всіх сегментах комп'ютерної індустрії - від мейнфреймів і мінікомп'ютерів до

персональних комп'ютерів і планшетних ПК. Програмні платформи також є основою на ринку смартфонів і стали ключовим фактором, за рахунок якого лідерами ринку стали iOS і Android. Програмні платформи також набувають все більшого значення в автомобільній промисловості і стають особливо важливі на ринку, що розвивається автомобілів, що залежать від програмного забезпечення.

У програмних платформах є і недоліки. Для їх реалізації потрібно написання великих обсягів коду, а працюють вони часто повільніше. Також вони можуть підвищувати вимоги до апаратної частини. Втім, в більшості випадків, переваги переважають недоліки [9].

Програмна платформа зазвичай являє собою набір з безлічі програмних модулів для реалізації певної функціональності. Втім, це можуть бути і невеликі програми, які використовуються в більших платформах. Таким чином, програмні платформи можуть значно відрізнитися за розмірами і складності коду. Ключовий аспект - перевикористання програмних модулів для економії витрат на розробку та обслуговування, а також підвищення рівня надійності за рахунок зниження загальної кількості помилок в ПЗ.

Ключем до використання і реалізації програмних платформ є API, що дозволяють створювати модульну структуру блоків і забезпечувати зв'язок між модулями або різними платформами. API конкретних сегментів ринку ПО, в разі успіху, створюють формують екосистему з взаємодоповнюючих програмних модулів, які з плином часу значно збільшують вартість і можливості використання платформи.

Програмні платформи на автомобільному ринку швидко ростуть і дуже впливають на всю індустрію автомобільного ПЗ, причому очікується, що в наступному десятилітті тенденція зростання збережеться.

Всі автомобільні ECU потребують програмного забезпечення, яке буде керувати різними апаратними компонентами, а також в додатках, які і будуть виконуватися блоком управління.

Всі автомобільні блоки управління потребують ПО, контролюючому роботу різноманітних програм, які керують апаратними компонентами і додатками, на які

розраховані всі електронні модулі керування. Консорціум автомобільної промисловості розробив і продовжує розширювати можливості платформи AUTOSAR. Ця технологія являє собою багаторівневу програмну архітектуру зі стандартними API-інтерфейсами для додатків, що виконуються ECU. Також ця платформа має незалежний інтерфейс, сумісний з більшістю мікропроцесорних архітектур, що використовуються в автомобільній промисловості. AUTOSAR - провідна програмна платформа в галузі систем допомоги водієві і інших традиційних додатках для ECU. Втім, AUTOSAR не може управляти блоками зі складною і великою кодовою базою так, як це роблять інформаційно-розважальні системи [10].

Все це дало можливість для операційних систем, які стали використовуватися в високотехнологічній промисловості в застосуванні до автомобільних систем. Операційні системи стали першими високотехнологічними програмними платформами і справили великий вплив на автомобільну промисловість. В сучасних інформаційно-розважальних і телематичних системах використовуються високотехнологічні ОС. Також складні операційні системи використовуються в ЕБУ, системи ADAS теж рухаються в цьому напрямку.

Деякі блоки управління мають підвищені вимоги до безпеки, і їм потрібні системи, що пройшли сертифікацію (наприклад, за стандартом ISO 26262). У теперішній момент, ні в одній з версій Linux немає таких сертифікатів. У таких блоках використовують QNX, Green Hills і інші аналогічні операційні системи, які мають сертифікати безпеки.

Мультимедійна інтерактивна системи – вищий рівень розвитку систем, що використовують ОС. На ринку платформ операційних систем йде конкуренція між QNX і різними версіями Linux - AGL, Android і GENIVI-сумісними версіями. Android від Google розвивається великими кроками. Успіх Android пов'язаний з основою Відкритого автомобільного альянсу (BAA) в 2014 році. В даний час в цей альянс входять 60 брендів, включаючи провідних OEM-виробників (Original equipment manufacturer) на зразок Chrysler, Fiat, Ford, GM, Honda, Hyundai, Mazda, Mercedes-Benz, Nissan, Renault, Toyota і VW.

У платформу ОС зазвичай вбудовані численні програми, які стосуються так званого проміжного програмного забезпечення, необхідного для забезпечення різноманітної функціональності. Проміжне ПЗ - свого роду програмний клей. Це програмне забезпечення надає різним додаткам доступ до сервісів, недоступним з ядра ОС. Проміжне ПЗ варіюється від утилітарних програм до апаратних інтерфейсів для автомобільних компонентів. НМІ і призначені для користувача інтерфейси також часто відносять до проміжного програмного забезпечення.

Гіпервізор - це невелика програмна платформа для управління декількома ОС і їх додатками.

Цю платформу також можна назвати монітором віртуальних машин (VMM). Віртуалізація використовується в комп'ютерній індустрії з 1960-х років і є ключовою технологією для центрів обробки даних. Гіпервізор набувають все більшого значення для автомобільного програмного забезпечення, оскільки вони вирішують цілий ряд проблем і можуть бути економічним рішенням.[12]

Можна навести два приклади, які демонструють переваги гіпервізора. Перший приклад: резервний монітор, який повинен обов'язково бути встановлений на всіх продаваних в США автомобілях є компонентом систем з підвищеними вимогами до безпеки. Другий приклад: дисплей в інформаційно-розважальних системах може бути ідеальним запасним засобом для виведення інформації, але при цьому в цій схемі не може використовуватися ОС Linux, оскільки у неї немає необхідних сертифікатів безпеки. Рішенням може стати гіпервізор, інтегруючий Linux з QNX, Green Hills або іншими ОС з сертифікатами безпеки.

Інтеграція кластера приладів та інформаційно-розважальної системи - ще один приклад ситуації, в якій гіпервізор є ефективним рішенням. Також Гіпервізор можуть застосовуватися в поєднанні з специфічними електронними блоками управління.

Телематичні системи мають вбудовані програмні платформи, які взаємодіють з SaaS-сервісами, що працюють з телематичними даними. У бортових телематичних системах використовується операційні системи, і лідером в цьому сегменті ринку є QNX (за рахунок того, що саме QNX використовується в OnStar).

Провідні постачальники першого рівня вбудовують в телематичні системи програмне і апаратне забезпечення. Деякі з Tier-1 постачальників інформаційно-розважальних систем також є провідними постачальниками телематичних систем.

Програмні платформи для телематичних систем на основі SaaS є власністю постачальника телематичних послуг (TSP) (таких як OnStar, SiriusXM, Verizon Telematics і WirelessCar). OnStar використовується тільки в автомобілях від GM, але у інших постачальників є цілий ряд OEM-клієнтів. Система WirelessCar в даний час в основному належить VW, та інвестування додаткових ресурсів, ймовірно, розширять його позиції на ринку.

Постачальники телематичних послуг починають використовувати хмарні ресурси Microsoft, AWS і Google - ці компанії впроваджують SaaS-платформи для функціональності мережових автомобілів.

Високотехнологічні інформаційно-розважальні системи - найскладніші автомобільні системи в серійному виробництві. Зазвичай в них вбудовані складні системи відображення зі звуковими і навігаційними системами, а також цілим рядом різних додатків [11].

Існують системи з можливістю підключення до телематичних систем, кластерам приладів і HUD. Інформаційно-розважальні системи також мають великий призначений для користувача інтерфейс для управління і взаємодії з водієм або пасажиром. У багатьох інформаційно-розважальних системах реалізовано підключення до програмної платформи для використання смартфонів і різних додатків в зв'язці з віртуальними особистими асистентами (VPA). Інформаційно-розважальні системи також активно використовують технології установки оновлень по повітрю і ПО для кібербезпеки.

Таким чином, в зв'язці з інформаційно-розважальною системою можуть використовуватися кілька різних програмних платформ. Ключовою програмною платформою в даній схемі є ОС, оскільки саме вона побічно керує всім ПЗ, пов'язаних з інформаційно-розважальною системою. У цих системах основну частку ринку займають різні операційні системи з групи Linux, зокрема дедалі

важливішою стає Android. Втім, вибір ОС для інформаційно-розважальної системи сильно варіюється в залежності від OEM-виробника.

Також у багатьох інформаційно-розважальних платформах використовуються навігаційні програмні системи - приблизно в 25% (частка залежить від країни). Навігаційні програмні платформи, як правило, поставляються компаніями, що спеціалізуються на навігації в автомобільній та інших промисловостях. Tier-1 Постачальники також можуть поставляти навігаційне програмне забезпечення.

У більшості сучасних інформаційно-розважальних систем використовуються дисплеї, і середній розмір дисплея зростає з кожним роком. В даний час дисплеї вбудовані приблизно в 60% інформаційно-розважальних систем. У деяких висококласних системах використовуються навіть графічні прискорювачі для підвищення продуктивності виведення на дисплей.

В даний час смартфон став невід'ємною частиною життя водія. Багато водіїв хочуть використовувати додатки зі своїх смартфонів в інформаційно-розважальних системах для того, щоб менше відволікатися на девайс. Деякі OEM-виробники розробили платформи для інтеграції мобільних додатків в інформаційно-розважальні системи. Втім, популярність CarPlay від Apple і Android Auto затьмарила всіх інших. Багато інформаційно-розважальні системи підтримують як CarPlay, так і Android Auto, щоб мати можливість взаємодіяти з більшістю сучасних смартфонів. Baidu CarLife використовується в основному в Китаї, і там у цієї системи дуже сильні позиції, оскільки Android Auto в Китаї недоступна [12].

Популярність CarPlay і Android Auto викликана двома основними факторами: знайомим призначенням для користувача інтерфейсом і великою кількістю специфічних для автомобілів додатків для обох платформ. Втім, додатки для iOS і Android не працюють як додатки для CarPlay або Android самі по собі. Ці додатки повинні бути модифіковані для роботи з певною платформою. Найбільш популярними є додатки з картами, даними про парковках і трафіку, музикою, подкастами, аудіокнигами, новинами, повідомленнями і календарем.

З CarPlay працюють більше 500 моделей автомобілів від 65 різних автовиробників. Android Auto працює в приблизно порівнянному кількості автомобілів від 60 автовиробників.

Голосові асистенти, вони ж віртуальні приватні асистенти (VPA) добре зарекомендували себе в автомобільній промисловості. Багато водіїв використовують голосових асистентів в своїх смартфонах або в домашніх пристроях. OEM-виробники намагалися створювати своїх асистентів, але високотехнологічні платформи з голосовими асистентами взяли верх. Ведучими розробниками голосових асистентів є ті ж компанії, що лідирують і на ринку розумних будинків і смартфонів: Amazon і їх Alexa, Google і їх асистент, Apple та Siri. Досягнення в галузі штучного інтелекту і технологій розпізнавання голосу сформували попит на голосових асистентів, оскільки вони допомагають водіям менше відволікатися [13].

Існує два способи використання голосових асистентів в автомобілі: через інтеграцію мобільних додатки в CarPlay або Android Auto або через окрему програмну платформу з асистентом в рамках автомобільної інформаційно-розважальної системи. Лідером в області інтегрованих асистентів є Alexa, асистент від Android тримає міцне друге місце. Siri від Apple в основному використовується в CarPlay, оскільки Apple є лідером на ринку інтеграції смартфонів. Alexa і асистент від Google також використовуються для інтеграції мобільних додатків.

Є інформація, що компанія OpenSynergy готує до випуску «автомобільну операційну систему» на базі Linux, а також материнську плату з процесором Intel Atom, призначену для побудови бортових інформаційно-розважальних систем. Платформа COQOS, за задумом розробників, повинна дозволити Linux-додаткам і AUTOSAR-модулям розділяти один і той же процесор класу SoC, ресурси якого будуть віртуалізувати за рахунок додаткового рівня «мікро операційної системи» (micro operating system, μ OS). AUTOSAR (AUTomotive Open System ARchitecture) - відкрита, стандартизована архітектура автомобільного програмного забезпечення, спільно розроблена автовиробниками, постачальниками компонентів і інструментів [14].

Згідно з інформацією, наведеною OpenSynergy, запропонована компанією технологія дозволяє розділяти між Linux-додатками і AUTOSAR-додатками рівні аудіо і графіки OpenGL, що, за словами виробника, вперше забезпечує можливість виведення на мультимедійні дисплеї і аудіосистеми з AUTOSAR-додатками. В якості додаткових переваг вказується скорочення кількості електронних блоків системи управління двигуном, зменшення вартості розробки, і навіть зниження витрати палива за рахунок скорочення різноманітності кабелів бортових мереж і, відповідно, їх ваги. Нещодавно було повідомлено про схожу ініціативу компанії Renesas Technology, яка розробила багатоядерний чіп SH7776, що дозволяє розділити ресурси автомобільних додатків на апаратному рівні. OpenSynergy пішла іншим шляхом - в її варіанті поділ віртуалізованих ресурсів забезпечується ОС реального часу μ OS. При цьому мікроядро μ OS виконує роль ОС AUTOSAR, наявність якої потрібно для виконання AUTOSAR-сумісних додатків. Взаємодія (interprocess communication, IPC) між AUTOSAR- і Linux-середовищами забезпечується за допомогою середовища часу виконання (runtime environments, RTE), тоді як Linux-завдання можуть взаємодіяти на рівні шини віртуальних додатків (Virtual Application Bus, VAB). Причому, за словами розробника, архітектура COQOS розрахована на підтримку як x86-сумісних, так і ARM-сумісних чіпів. OpenSynergy повідомила, що розробка першої версії COQOS і тестової плати на базі Intel Atom близька до завершення.

3.3 ДИСТАНЦІЙНЕ ОНОВЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

З бурхливим розвитком автомобільних електронних технологій, складність та кількість програм управління автомобілем в ЕБУ збільшуються, проблеми з гарантією та відкликанням автомобіля, спричинені програмним кодом, відповідно стають все більш частими, традиційний спосіб оновлення ЕБУ автомобіля важко задовольнити вимоги розвитку автомобільної електронної техніки. Тому пропонується реалізація програми дистанційного бездротового оновлення автомобільних ЕБУ, яка завантажує файли оновлень з інформаційної платформи

автомобіля на віддалений сервер і передає їх до автомобільного ECU. Було вивчено проблеми безпеки, які можуть існувати в системі віддаленого оновлення, та запропонували відповідні рішення. На основі цієї схеми в роботі було розглянуто конфігурацію віддаленого сервера, конструкцію платформи автомобіля та дизайн завантажувача електронного блоку управління автомобіля, а також завершено розробку програми всього процесу віддаленого оновлення ЕБУ. Нарешті, перевіряється можливість використання системи. Загалом, у майбутньому, коли транспортні засоби потребуватимуть відкликання через проблеми програмного коду, можна звернутися до системи віддаленого оновлення, що розробляється, встановити відповідний механізм оновлення, щоб спростити процес оновлення ЕБУ, забезпечити безпеку оновлення електронного модулю керування та заощадити витрати часу власника та виробників хостів.

Пакет оновлення надсилається віддаленим сервером, а отримується бортовим програмним та апаратним забезпеченням. У цьому процесі має бути виконано шість функцій: завантаження пакета оновлення програмного забезпечення електронного блоку керування, управління інформацією про версію програмного забезпечення, випуск та завантаження пакету оновлення, бездротове підключення до мережі, управління безпекою та оновлення програмного забезпечення ЕБУ.

Зокрема, на рисунку 1 показано телекомунікаційний механізм, який можна узагальнити за 7 кроків. Виробники або постачальники завантажують оновлення мікропрограмного забезпечення ЕБУ на сервер, клієнт платформи транспортного засобу перевіряє, чи доступний пакет оновлення програмного забезпечення ЕБУ під час запуску автомобіля. Якщо доступне програмне забезпечення ЕБУ є новим пакетом, платформа автомобіля повідомляє користувача через спливаюче повідомлення, яке відображається системою. Відповідно до спливаючого вікна повідомлення, користувачі вибирають операцію оновлення, наприклад, негайне або відстрочене оновлення. Якщо користувач підтверджує операцію оновлення, платформа транспортного засобу надсилає запит на сервер для завантаження пакету оновлення та збереження на локальному запам'ятовуючому пристрої, або чекає наступного раунду повідомлення. Отримавши запит на завантаження, сервер

надсилає пакети оновлення програмного забезпечення до автомобіля за допомогою бездротової технології 4G / Wifi. Після завершення завантаження сервер надсилає успішно завантажену інформацію на платформу автомобіля. Клієнт платформи транспортного засобу встановлює пакет оновлення програмного забезпечення ЕБУ через мережу CAN транспортного засобу. Після автоматичного оновлення платформа автомобіля надсилає на сервер код транспортного засобу та оновлення інформації про успіх [15].

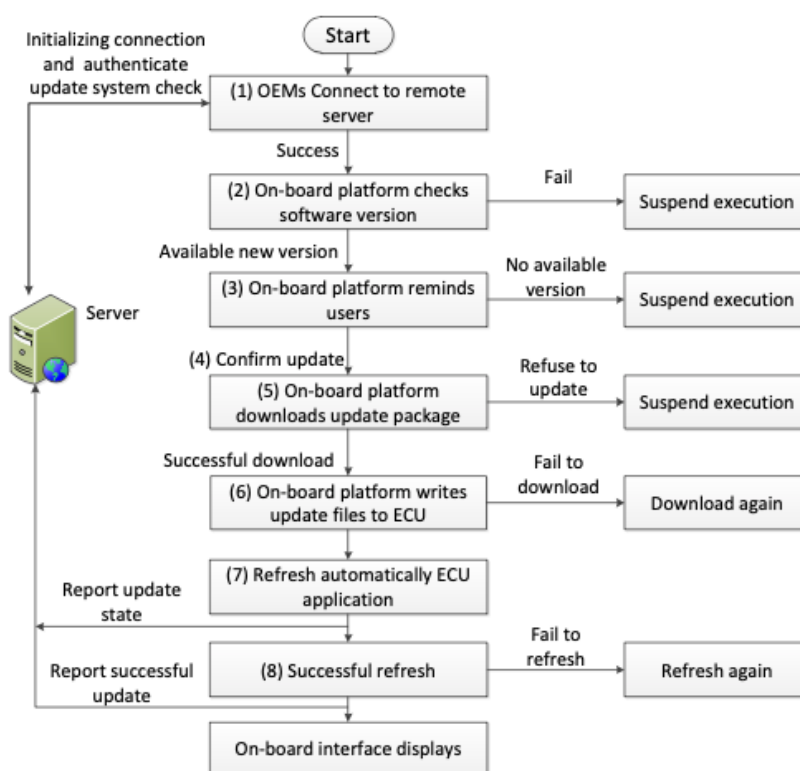


Рисунок 2 – Процес оновлення програмного забезпечення

3.4 ДОСЛІДЖЕННЯ СИСТЕМ БЕЗПЕКИ ЕЛЕКТРОННОГО МОДУЛЯ КЕРУВАННЯ

Під час хакерської конференції 2015 року DEFCON було атаковано Jeep, Tesla Model S та інші транспортні засоби, що є сигналом того, що питання безпеки автомобільної мережі стають все більш важливими.

Загалом, мережа транспортних засобів повинна мати кілька функцій безпеки: автентифікація особи, виявлення ненормальної поведінки, цілісність даних, конфіденційність даних. Однак у системі віддаленого оновлення існують

потенційні проблеми, такі як зворотний розвиток файлів S19, втручання у файли S19, завантаження ненадійного програмного забезпечення та незаконний доступ до ЕБУ, які не можуть бути ефективно вирішені загальними мережевими функціями. Таким чином, цей документ покращить загальну схему віддаленого оновлення та додасть відповідний механізм безпеки для слабких вузлів.

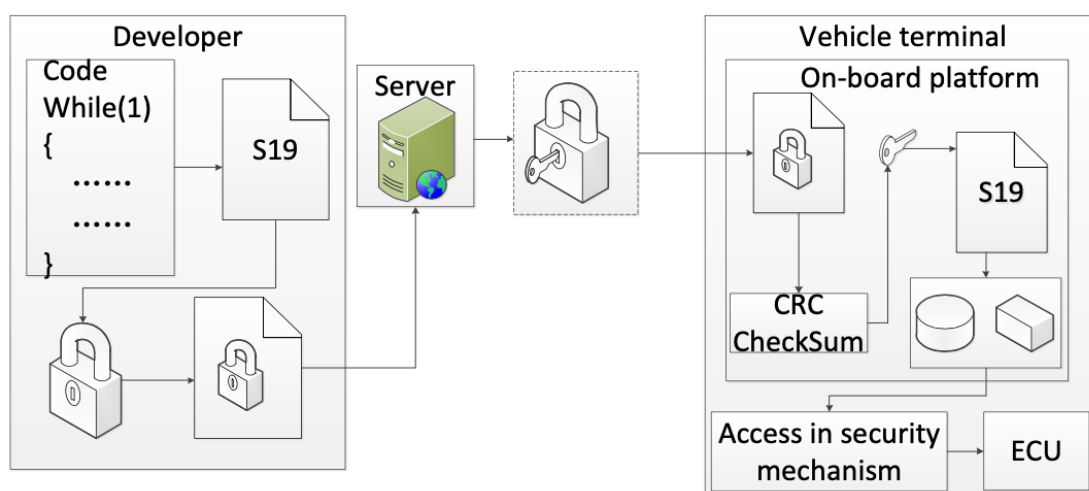


Рисунок 3 – Віддалене оновлення ПЗ з системою безпеки

Коли справа стосується проблеми зворотного розвитку файлу S19, файл може бути зашифрований, а саме виробники та постачальники спочатку шифрують файл S19, а потім завантажують його на віддалений сервер. Після отримання зашифрованого оновлення платформа автомобіля розшифровується, щоб отримати файли S19. Тож навіть файл отримують не уповноважені органи, вони не можуть отримати вміст. Для проблеми, в якій файл підроблений, додається перевірка цілісності даних, щоб платформа транспортного засобу могла точно визначити, чи файл змінено, а потім визначити, чи повинен діяти наступний крок. Додавши механізм автентифікації ідентифікаційних даних на основі інформації про сертифікат програмного пакету, платформа автомобіля може перевірити джерело пакету та ефективно запобігти входу невіреного пакету оновлення програмного забезпечення до платформи транспортного засобу. Після додавання механізму доступу до захищеного ECU термінал, суміщений з механізмом шифрування ECU,

може увійти в ECU для його оновлення. Конкретний процес показаний на рисунку 3.

Посилання для шифрування пакетів оновлення використовує алгоритм шифрування AES, який є симетричним алгоритмом шифрування. AES може бути використаний для шифрування та дешифрування 128-бітової групи даних. Перевага симетричного методу шифрування полягає в тому, що він може призвести до менших обчислень і високої ефективності обчислення, використовуючи прості логічні операції, такі як AND, OR, XOR для шифрування та дешифрування. Для віддаленого оновлення пакетного шифрування з великим обсягом даних і довгим потоком даних для ефективності більш доцільним є використання симетричного алгоритму шифрування. Процес віддаленого оновлення пакетного шифрування полягає в наступному. По-перше, файл S19 потрібно розділити на N-груп. Кожна група складає 128 біт, а саме 16 байт. Однак іноді файл S19 не можна просто розділити на групи з 128-біт. У таких випадках перед групуванням його потрібно заповнити після аналізу даних файлу S19, щоб він міг задовольнити потреби групування 128 бітів. Як показано на малюнку 3, на основі N відкритого тексту пакета даних алгоритм шифрування AES і 128-бітний ключ використовуються для отримання n груп тексту шифру, а саме зашифрованого файлу S19. У цій роботі 128-бітний алгоритм шифрування ключа використовується для шифрування та дешифрування файлів S19 у 10 раундів. Процес дешифрування алгоритму симетричного шифрування насправді є інверсією процесу шифрування.

Алгоритм AES - це алгоритм шифрування групування, що означає, що відкритий текст для кожної групи даних шифрується окремо. У процесі передачі, якщо набір даних було знищено або підроблено, це важко виявити. Тому платформа автомобіля також повинна оновити пакет для перевірки цілісності даних.

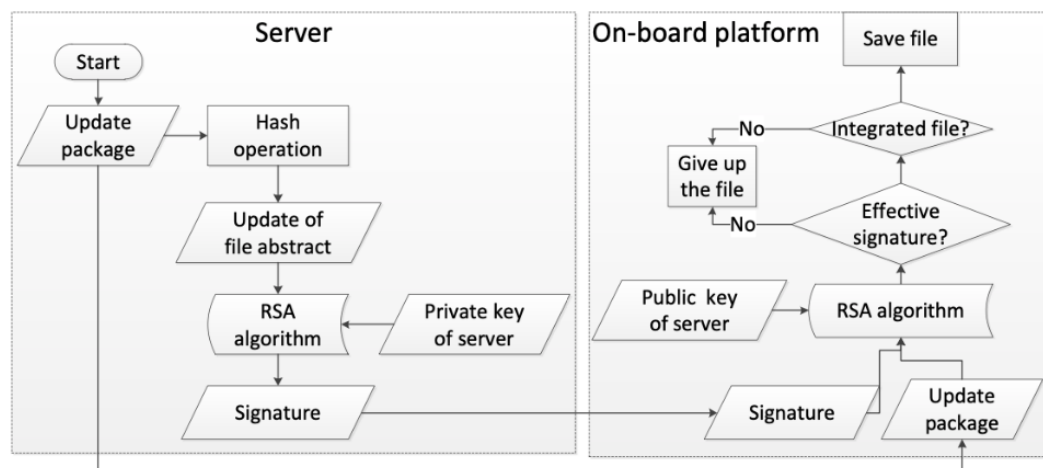


Рисунок 4 – Аутентифікація та перевірка цілісності ПЗ. Блох-схема алгоритму.

Аутентифікація для пакету оновлення потрібна головним чином для забезпечення надійності його джерел та для запобігання ненадійним установам надсилати пакет оновлення на платформу автомобіля. Перевірка цілісності даних дозволяє перевірити, чи були дані підроблені чи ні, і переконатися, що файл, завантажений на платформу автомобіля, є ефективним пакетом оновлення ECU. На рисунку 5 показано процес автентифікації та перевірки цілісності пакету оновлення на основі алгоритму цифрового підпису RSA [16].

Сторона сервера: по-перше, пакет оновлень хешований, щоб сформувати анотацію інформації про оновлений файл. Потім анотація та секретний ключ сервера використовуються для генерації підпису алгоритмом генерації підписів RSA. Нарешті, підпис та пакет оновлень надсилаються на платформу автомобіля.

Сторона платформи транспортного засобу: після отримання пакету оновлення та підпису від сервера, відкритий ключ сервера застосовується для перевірки через алгоритм RSA. Дійсний підпис означає, що пакет оновлення є приватним ключем сервера, який є надійним джерелом. Потім її цілісність додатково перевіряється. Якщо він не пошкоджений, файл буде збережено.

Відповідно до положень розділу оновлення ЕБУ в діагностичному протоколі LAN контролю дорожнього руху (ISO 15765), безпечний доступ вказаний як рекомендований етап оновлення ЕБУ автомобіля. Положення протоколу загальної

діагностики поломок транспортного засобу ISO 14299 (ISO 14299) про службу безпеки забезпечують базовий процес оновлення ЕБУ. Механізм захисту доступу в цьому документі базується на ISO 14299, як показано на рисунку 5.

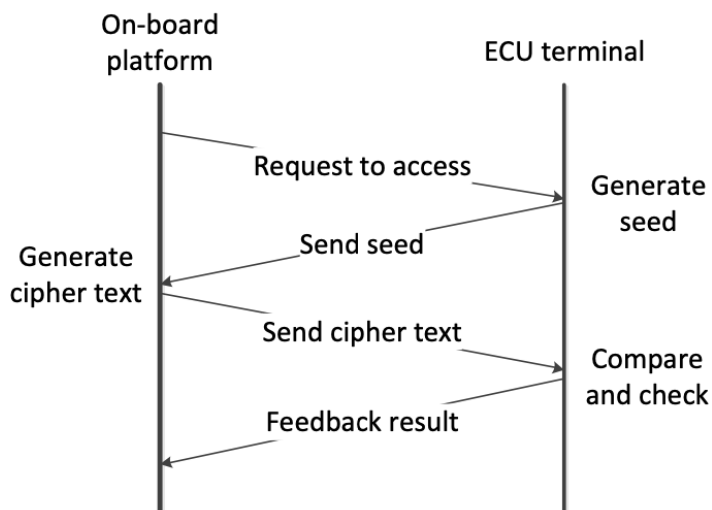


Рисунок 5 – Процес перевірки безпеки.

Включаються наступні кроки:

1. Комп'ютер платформи автомобіля запитує доступ до ЕБУ;
2. Термінал ЕБУ транспортного засобу генерує випадкове ядро і відправляє його на платформу транспортного засобу. У той же час ЕБУ використовує ключ та код як вхідні дані для обчислення тексту шифру за узгодженим алгоритмом шифрування;
3. Комп'ютер платформи транспортного засобу приймає програмний код. Потім програмне забезпечення шифрується за тим самим алгоритмом та ключем, щоб отримати текст шифру. Далі текст шифру надсилатиметься в ЕБУ автомобіля;
4. ECU порівнює текст шифру з платформи автомобіля та текст шифру, сформований у кроці 2. Узгоджені ключі означають, що запитуючий пристрій має узгоджений ключ і знає алгоритм шифрування ключа, термінал ECU є законним, тому ECU буде розблоковано. В іншому випадку ЕБУ заборонить доступ;
5. Результат надсилається на платформу транспортного засобу.

3.5 РЕАЛІЗАЦІЯ ФУНКЦІЙ ПЛАТФОРМИ АВТОМОБІЛЯ ТА РОЗРОБКА ДОДАТКІВ

Операційну систему Android обрано для використання на платформі автомобіля, враховуючи її відкритий код, велику кількість програм та низькі витрати на розробку. За цією умовою основним процесорним чіпом для плати розробки є i.MX 6Quad, який має чотири ядра. Його робоча частота до 1,2 ГГц. Пристрої цієї серії інтегрують FlexCAN, шину MLB, PCI Express, порт дисплея MIPI, інтерфейс порту камери MIPI тощо. Іншими словами, це прикладна платформа для побутової електроніки, автомобільної та промислової мультимедіа.

Конкретний процес перенесення андроїда [17] наведений на рисунку 7. По-перше, слід створити середовище для розробки. Потрібно отримати вихідний код Android. Враховуючи, що кількість вихідного коду Android дуже велика, тому його потрібно розділити на дві частини, щоб отримати, відповідно, вихідний код ядра Android та Android (ядро Android Linux). Тоді ядро Linux та файлова система Android слід скомпілювати окремо. Нарешті файл зображення, створений компілятором, завантажується на апаратну платформу. Функція програми на платформі Android полягає в завантаженні пакета оновлення ECU віддаленого сервера на платформу автомобіля, а потім платформа автомобіля завантажує файл оновлення ECU в ECU автомобіля через шину CAN.

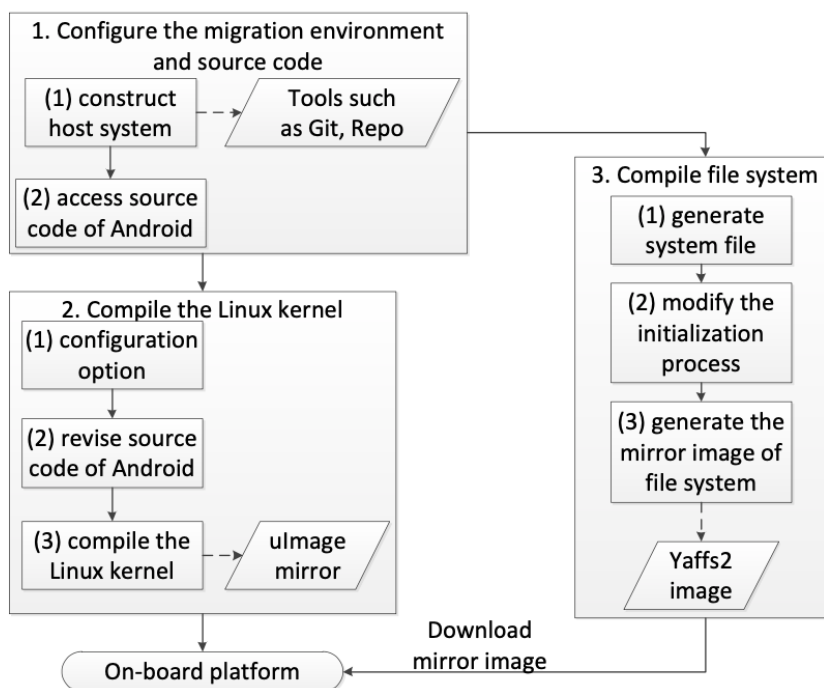


Рисунок 6 – Процес перенесення Android.

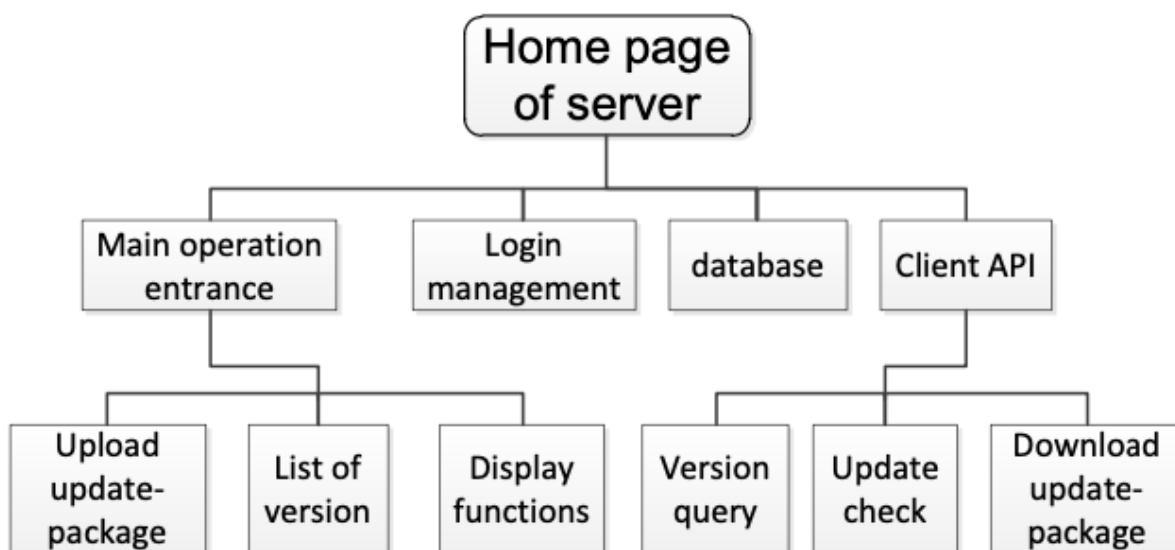


Рисунок 7 – Дерево додатків на сервері.

Функції віддаленого сервера оновлення включають вхід користувача, віддалене завантаження пакету оновлення, перевірку оновлення ЕБУ, перевірку безпеки та оновлення ЕБУ. Функція завантаження пакету оновлення ЕБУ для системи віддаленого оновлення в основному призначена для користувачів виробників та постачальників, а не власників автомобілів. Коли у автомобіля виходить з ладу ЕБУ або потребує оновлення програмного забезпечення,

розробники додатків ЕБУ повинні завантажити пакет оновлення на віддалений сервер, щоб платформа автомобіля могла завантажити пакет оновлень для оновлення програмного забезпечення ЕБУ. Функція перевірки оновлення ЕБУ призначена для платформи автомобіля, щоб перевірити, чи є на сервері новий пакет для оновлення чи ні. Перевірка оновлення досягається головним чином за рахунок порівняння відповідної інформації між віддаленою службою оновлення та пакетом оновлення ЕБУ.

Розробка додатків на борту. Додатки розроблені з метою впровадження п'яти функцій дистанційного оновлення.

1. Вхід користувача платформи транспортного засобу.

Отримавши інформацію про пароль вхідного облікового запису, клієнт платформи транспортного засобу підключається до входу на сервер для перевірки порту та надсилає інформацію про пароль облікового запису на сервер. Основний код такий:

```
httpsRequest.execute(Common.ServerIp+"/session/logincheck.php","username",username.getText().toString(), "password",password.getText().toString());
```

2. Перевірка оновлення ECU

За умови перевірки оновлення, сторона клієнта пов'язана з “update_check” модуля інтерфейсу клієнтського API віддаленого сервера для оновлення інтерфейсу перевірки через коди:

```
mHttpsAsyncRequest.execute(Common.ServerIp +  
"/client_api/update_check.php")
```

3. Завантажити пакет оновлення

Етап спілкування клієнта серверної та автомобільної платформ починається з клієнта платформи автомобіля поширюється на сервер для завантаження програми, закінчуючи завантаженням програмного забезпечення пакета ЕБУ автомобіля на SD-карту платформи автомобіля у зазначеній папці.

4. Оновлення ECU

Процес оновлення включає чотири основних етапи: завантаження пакету оновлення; оновити розбір пакета; стирання блоку спалаху; написання флеш-блоку. Завантаження пакету оновлення означає отримання даних файлів оновлення S19 з бортової платформи через шину CAN. Файли S19 потрібно проаналізувати для запису на Flash. Однак спочатку Flash із вихідними кодовими даними слід стерти, а потім записати.

Розробка завантажувача терміналу ECU автомобіля. У цій системі віддаленого оновлення програма завантажувача на ЕБУ головним чином відповідає за завантаження пакету оновлень з платформи автомобіля у відповідну пам'ять ЕКЮ, щоб досягти оновлення програмного забезпечення ЕБУ в режимі онлайн.

Документи S19 мають в основному три типи записів S0, S1 і S9. S0 являє собою початок програми, тоді як S9 являє собою кінець програми. Два типи не потрібно записувати на Flash. Адресне поле в записі S1 представлено 2 байтами, поле даних складається з даних, які можна завантажити в пам'ять, яка є основним вмістом файлу S19. Всі коди програми оновлення записуються записом S1. Таким чином, суть оновлення ECU полягає у записі поля даних запису S1 до блоку пам'яті Flash, який задається полем адреси. Візьмемо як приклад функцію запису та стирання P-Flash: визначено масив структури для збереження вмісту кожного запису у файлі.

```
typedef struct SRecord {
    uint8 Тип; // номер запису
    uint8 DataLength; // довжина даних
    uint32 LoadAddr; // початкове значення адреси
    uint8 Дані [64]; // масив поля даних
    unit8 CheckSum // контрольна сума даних
} * SRecData;
```

Згідно з вищевказаними кроками, функція стирання сектору P-Flash, а саме PFlash_EraseSector (адреса UINT32) та функція запису P-Flash, а саме

PFlash_Program (адреса UINT32, * ptr UINT16) призначені для реалізації операції стирання та запису Flash. Використовуючи ці дві функції з даними, проаналізованими з файлу S19, через процес, показаний на рисунку 8, обхід структури даних через пакет оновлення та витяг відповідного коду адрес та коду даних. По-перше, слід використовувати функцію PFlash_EraseSector, щоб стерти сектор P-Flash, що відповідає інформації про адресу, а потім записати відповідні дані коду в сектор через PFlash_Program, прочитавши наступну перевірку запису після перевірки та виправити до кінця запису S9, потім завершити операцію запису файлів оновлення.

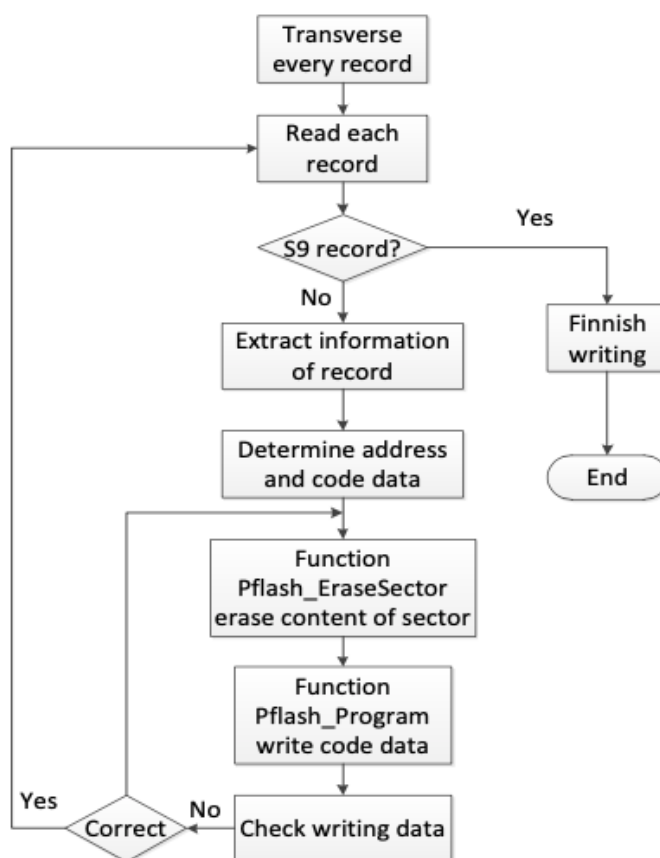


Рисунок 8 – Процес завантаження пакету оновлення сервера. Блок-схема алгоритму.

3.6 ЕКСПЕРИМЕНТАЛЬНА ПЕРЕВІРКА

Основним робочим процесом системи дистанційного оновлення автомобіля ЕБУ, розробленої в цій роботі, є: по-перше, розробники повинні оновити зашифровані файли оновлення ЕБУ, завантажені на віддалений сервер; платформа автомобіля завантажує пакет оновлення з віддаленого сервера та розшифровує його, щоб отримати файл S19; клієнт платформи автомобіля за допомогою функції оновлення ЕБУ оновить файл S19 до ЕБУ; функції також потрібно використовувати завантажувач ЕБУ, щоб отримати файл оновлення з платформи автомобіля та записати його в пам'ять ЕБУ, завершивши операцію віддаленого оновлення.

Віддалена служба побудована на архітектурі сервера LAMP Linux; платформа автомобіля - це платформа розробки системи Android, заснована на iMX6, що підключається до віддаленого сервера за допомогою WIFI / 3G, використовуючи шину CAN для зв'язку з автомобільним ECU; автомобільний ЕБУ використовує мікропроцесорну плату Freescale MC9S12XS128 для проведення моделювання.

Проводяться три кроки для перевірки роботи терміналу автомобіля, перший - це завантаження мікропрограми, перевірка локального пакета мікропрограмного забезпечення, нарешті, впровадження операційної системи оновлення та спостереження за тим, чи успішно оновлено автомобільний ЕБУ.

Після здійснення відповідної операції, дотримуючись послідовності світлодіодних вогнів на ЕБУ автомобіля. Це показує, що програма освітлення запущена, що означає, що вся операція віддаленого оновлення успішно завершена.

Висновки до розділу 3

В розділі 3 було розглянуто передачу даних між електронними модулями керування автомобіля - технологія Autosar. Технологія Autosar містить всі необхідні формати обміну і шаблони, які використовуються як для генерації і конфігурації інфраструктури, так і для її опису.

Інформація аналізується з метою створення об'єктивної моделі навколишнього середовища, що дозволяє формувати нові опції, які підтримують водія в екстремальних випадках.

Autosar переслідує мету створити та встановити відкриту та стандартизовану архітектуру програмного забезпечення для автомобільних електронних блоків управління (ЕБУ).

Досліджено систему віддаленого оновлення програмного забезпечення. Пакет оновлення надсилається віддаленим сервером, а отримується бортовим програмним та апаратним забезпеченням.

4. РЕАЛІЗАЦІЯ АДАПТИВНОГО АЛГОРИТМА

4.1 ОСНОВНІ ПРИНЦИПИ АДАПТИВНОГО АЛГОРИТМА

Алгоритм, що розробляється в даній магістерській дисертації базується на 5 принципах:

1. Аналіз. Алгоритм буде в першу чергу ґрунтуватися на аналізі всіх факторів, що можуть вплинути на потужність автомобіля. В першу чергу це якість пального, температура та щільність повітря, що втягується в двигун автомобіля.

Полюова шина CAN здатна передавати дані зі швидкістю 1 Мбіт/с. Це дозволяє передавати дані з великою швидкістю для аналізу та корегування заданих параметрів роботи.

Промислова мережа реального часу CAN є мережею із загальною середовищем передачі даних. Це означає, що всі вузли мережі одночасно приймають сигнали передаються по шині. Неможливо послати повідомлення будь-якого конкретного вузла. Всі вузли мережі приймають весь трафік, що передається по шині. Однак, CAN-контролери надають апаратну можливість фільтрації CAN-повідомлень.

Кожен вузол складається з двох складових. Це власне CAN контролер, який забезпечує взаємодію з мережею і реалізує протокол, і мікропроцесор (CPU).[18]

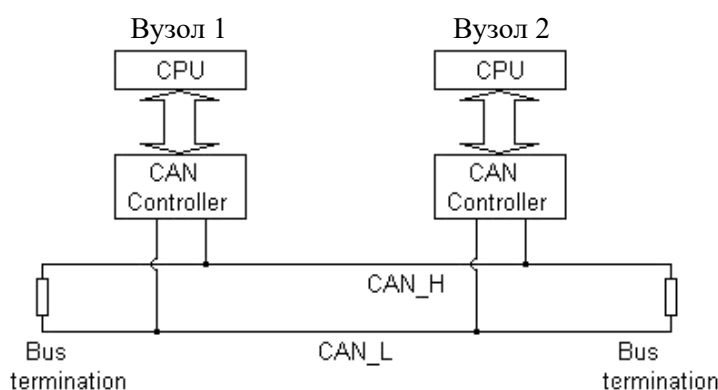


Рисунок 9 – Топологія мережі CAN

CAN контролери з'єднуються за допомогою диференціальної шини, яка має дві лінії - CAN_H (can-high) і CAN_L (can-low), по яких передаються сигнали. Логічний нуль реєструється, коли на лінії CAN_H сигнал вище, ніж на лінії CAN_L. Логічна одиниця - в разі коли сигнали CAN_H і CAN_L однакові (відрізняються менш ніж на 0.5 В). Використання такої диференціальної схеми передачі робить можливим роботу CAN мережі в дуже складних зовнішніх умовах. Логічний нуль - називається домінантним бітом, а логічна одиниця - рецесивним. Ці назви відображають пріоритет логічної одиниці і нуля на шині CAN. При одночасній передачі в шину логічного нуля і одиниці, на шині буде зареєстрований тільки логічний нуль (домінантний сигнал), а логічна одиниця буде подавлена (рецесивний сигнал).

Дані в CAN передаються короткими повідомленнями-кадрами стандартного формату. У CAN існують чотири типи повідомлень:

- Data Frame
- Remote Frame
- Error Frame
- Overload Frame

Data Frame - це найбільш часто використовуваний тип повідомлення. Він складається з наступних основних частин:

- поле арбітражу (arbitration field) визначає пріоритет повідомлення в разі, коли два або більше вузлів одночасно намагаються передати дані в мережу. Поле арбітражу полягає в свою чергу з:
 - для стандарту CAN-2.0A, 11-бітного ідентифікатора + 1 біт RTR (retransmit)
 - для стандарту CAN-2.0B, 29-бітного ідентифікатора + 1 біт RTR (retransmit)

Слід зазначити, що поле ідентифікатора, незважаючи на свою назву ніяк не ідентифікує саме по собі ні вузол в мережі, ні вміст поля даних. Для Data кадру біт RTR завжди виставлений в логічний нуль (домінантний сигнал).

- поле даних (data field) містить від 0 до 8 байт даних
- поле CRC (CRC field) містить 15-бітну контрольну суму повідомлення, яка використовується для виявлення помилок
- слот підтвердження (Acknowledgement Slot) (1 біт), кожен CAN-контролер, який правильно прийняв повідомлення посилає біт підтвердження в мережу. Вузол, який послав повідомлення, оброблює цей біт, і в разі, якщо підтвердження не прийшло, повторює передачу. У разі прийому слота підтвердження передавальний вузол може бути впевнений лише в тому, що хоча б один з вузлів в мережі правильно прийняв його повідомлення.

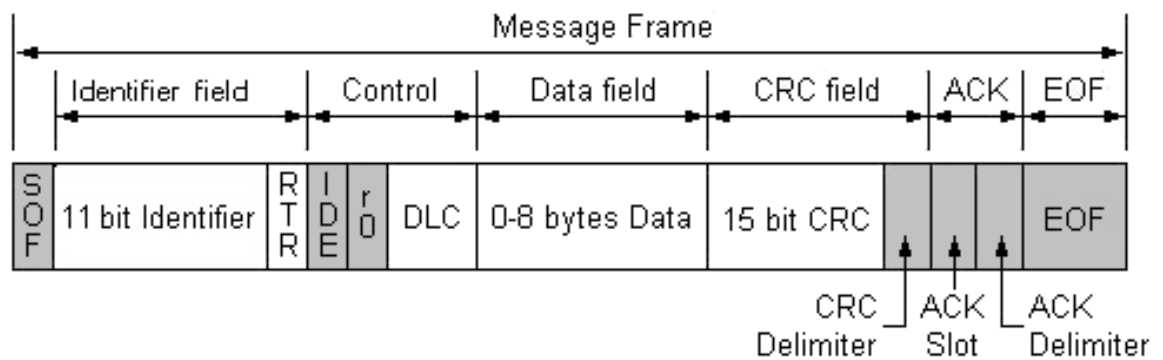


Рисунок 10 – Data frame стандарту CAN 2.0A.

Remote Frame - це Data Frame без поля даних і з виставленим бітом RTR (1 - рецесивні біт). Основне призначення Remote кадру - це ініціація одним з вузлів мережі передачі в мережу даних іншим вузлом. Така схема дозволяє зменшити сумарний трафік мережі. Однак, на практиці Remote Frame зараз використовується рідко (наприклад, в DeviceNet Remote Frame зовсім не використовується).

Error Frame - це повідомлення яке явно порушує формат повідомлення CAN. Передача такого повідомлення призводить до того, що всі вузли мережі реєструють помилку формату CAN-кадру, і в свою чергу автоматично передають в мережу Error Frame. Результатом цього процесу є автоматична повторна передача даних в мережу передавальним вузлом. Error Frame складається з поля Error Flag, яке складається з 6 біт однакового значення (і таким чином Error frame порушує

перевірку Bit Stuffing, див. Нижче), і поля Error Delimiter, що складається з 8 рецесивних бітів. Error Delimiter дає можливість іншим вузлам мережі виявивши Error Frame послати в мережу свій Error Flag.

Overload Frame - повторює структуру і логіку роботи Error кадру, з тією різницею, що він використовується перевантаженим вузлом, який в даний момент не може обробити повідомлення, яке надходить, і тому просить за допомогою Overload-кадру про повторну передачу даних. В даний час Overload-кадр практично не використовується.

Контроль доступу до середовища передачі (побітовий арбітраж).

Поле арбітражу CAN-кадру використовується в CAN для вирішення колізій доступу до шини метод неструктивного арбітражу. Суть методу неструктивного арбітражу полягає в наступному. У разі, коли кілька контролерів починають одночасну передачу CAN кадру в мережу, кожен з них порівнює біт, який збирається передати на шину, з бітом, який намагається передати на шину конкуруючий контролер. Якщо значення цих бітів рівні, обидва контролера передають наступний біт. І так відбувається до тих пір, поки значення переданих бітів не стануть різними. Тепер контролер, який передавав логічний нуль (більш пріоритетний сигнал) буде продовжувати передачу, а інший (інші) контролер обірве свою передачу до того часу, поки шина знов не звільниться. Звичайно, якщо шина в даний момент зайнята, то контролер не почне передачу до моменту її звільнення.

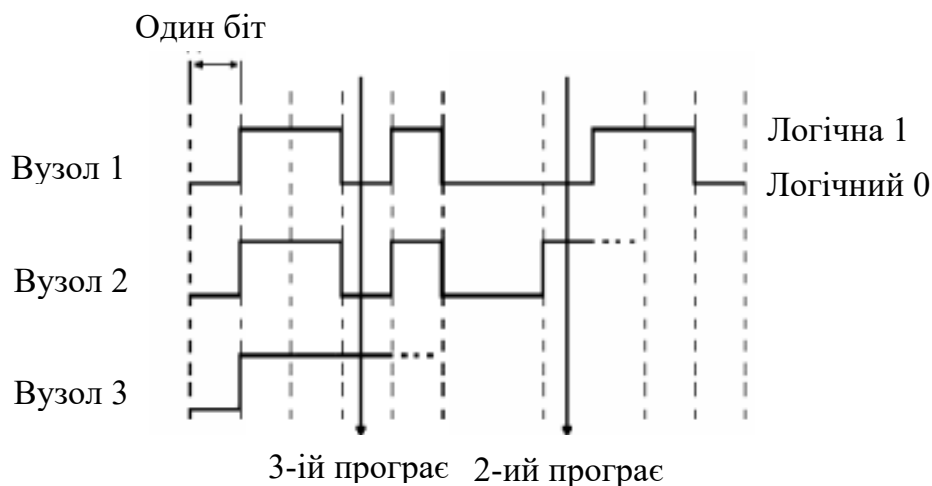


Рисунок 11 – Побітовий арбітраж на шині CAN.

Методи виявлення помилок.

CAN протокол визначає п'ять способів виявлення помилок в мережі:

Bit monitoring

Bit stuffing

Frame check

ACKnowledgement Check

CRC Check

Bit monitoring - кожен вузол під час передачі бітів в мережу порівнює значення переданого ним біта зі значенням біта, яке з'являється на шині. Якщо ці значення не збігаються, то вузол генерує помилку Bit Error. Природно, що під час арбітражу на шині (передача поля арбітражу в шину) цей механізм перевірки помилок відключається.

Bit stuffing - коли вузол передає послідовно в шину 5 біт з однаковим значенням, то він додає шостий біт з протилежним значенням. Приймаючі вузли цей додатковий біт видаляють. Якщо вузол виявляє на шині більше 5 послідовних біт з однаковим значенням, то він генерує помилку Stuff Error.

Frame Check - деякі частини CAN-повідомлення мають однакове значення у всіх типах повідомлень. Тобто протокол CAN точно визначає які рівні напруги і коли повинні з'являтися на шині. Якщо формат повідомлень порушується, то вузли генерують помилку Form Error.

ACKnowledgement Check - кожен вузол отримавши правильне повідомлення по мережі посилає в мережу домінантний (0) біт. Якщо ж цього не відбувається, то передавальний вузол реєструє помилку Acknowledgement Error.

CRC Check - кожне повідомлення CAN містить CRC суму, і кожне приймаюче вузол підраховує значення CRC для кожного отриманого повідомлення. Якщо підрахована значення CRC суми, не збігається зі значенням CRC в тексті листа, який приймає вузол генерує помилку CRC Error.

Механізм обмеження помилок (Error confinement).

Кожен вузол мережі CAN, під час роботи намагається виявити одну з п'яти можливих помилок. Якщо помилку виявлено, вузол передає в мережу Error Frame, руйнуючи тим самим весь поточний трафік мережі (передачу і прийом поточного повідомлення). Всі інші вузли виявляють Error Frame і приймають відповідні дії (скидають прийняте повідомлення). Крім того, кожен вузол веде два лічильники помилок: Transmit Error Counter (лічильник помилок передачі) і Receive Error Counter (лічильник помилок прийому). Ці лічильники збільшуються або зменшуються в відповідність з декількома правилами. Самі правила управління лічильниками помилок досить складні, але зводяться до простого принципу, помилка передачі призводить до збільшення Transmit Error лічильника на 8, помилка прийому збільшує лічильник Receive Error на 1, будь-яка коректна передача / прийом повідомлення зменшуються відповідний лічильник на 1. Ці правила призводять до тому, що лічильник помилок передачі передавального вузла збільшується швидше, ніж лічильник помилок прийому приймають вузлів. Це правило відповідає припущенням про велику ймовірність того, що джерелом помилок є передавальний вузол.

Кожен вузол CAN мережі може знаходитися в одному з трьох станів. Коли вузол стартує, він знаходиться в стані Error Active. Коли значення хоча б одного з двох лічильників помилок перевищує межу 127, вузол переходить в стан Error Passive. Коли значення хоча б одного з двох лічильників перевищує межу 255, вузол переходить в стан Bus Off.

Вузол знаходиться в стані Error Active, в разі виявлення помилки на шині передає в мережу Active Error Flags. Active Error Flags сотстоїт з 6 доміантних біт, тому всі вузли його реєструють. Вузол в стані Error Passive передає в мережу Error Passive Flags при виявленні помилки в мережі. Error Passive Flags складається з 6 рецесивних біт, тому інші вузли мережі його не помічають, і Error Passive Flags лише призводить до збільшення Error лічильника вузла. Вузол в стані Bus Off нічого не передає в мережу (не тільки Error кадри, але взагалі ніякі інші).

Адресація і протоколи високого рівня.

У CAN не існує явної адресації повідомлень і вузлів. Протокол CAN ніде не вказує що поле арбітражу (Identification field + RTR) має використовуватися як ідентифікатор повідомлення або вузла. Таким чином, ідентифікатори повідомлень і адреси вузлів можуть знаходитися в будь-якому полі повідомлення (в поле арбітражу або в поле даних, або бути присутнім і там, і там). Точно також протокол не забороняє використовувати поле арбітражу для передачі даних.

Утилізація поля арбітражу і поля даних, і розподіл адрес вузлів, ідентифікаторів повідомлень і пріоритетів в мережі є предметом розглядів так званих протоколів високого рівня (HLP - Higher Layer Protocols). Назва HLP відображає той факт, що протокол CAN описує тільки два нижніх рівні еталонної мережевої моделі ISO / OSI, а інші рівні описуються протоколами HLP.

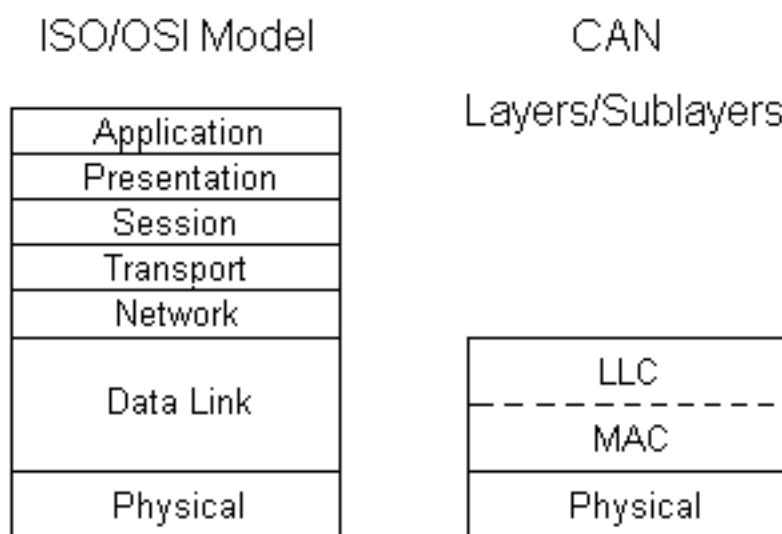


Рисунок 12 – Логічна структура протоколу CAN.

Фізичний РІВЕНЬ Протоколу CAN

Фізичний рівень (Physical Layer) протоколу CAN визначає опір кабелю, рівень електричних сигналів в мережі і т.п. Існує кілька фізичних рівнів протоколу CAN (ISO 11898, ISO 11519, SAE J2411).

У переважній більшості випадків використовується фізичний рівень CAN певний в стандарті ISO 11898. ISO 11898 в якості середовища передачі визначає

двопровідну диференціальну лінію з опором (термінатори) 120 Ом (допускається коливання імпедансу в межах від 108 Ом до 132 Ом. Фізичний рівень CAN реалізований в спеціальних чіпах - CAN приймально-передавачах (transceivers), які перетворюють звичайні TTL рівні сигналів використовуваних CAN-контролерами в рівні сигналів на шині CAN. Найбільш поширений CAN приймально-передавач - Phillips 82C250, який повністю відповідає стандарту ISO 11898.

Максимальна швидкість мережі CAN у відповідність з протоколом дорівнює 1 Mbit / sec. При швидкості в 1 Mbit / sec максимальна довжина кабелю дорівнює приблизно 40 м. Обмеження на довжину кабелю пов'язано з кінцевою швидкістю світла і механізмом побітового арбітражу (під час арбітражу всі вузли мережі повинні отримувати поточний біт передачі одночасно, ті сигнал повинен встигнути поширитися по всьому кабелю за одиничний відлік часу в мережі. Співвідношення між швидкістю передачі і максимальною довжиною кабелю наведено в таблиці:

Таблиця 1 – Залежність швидкості передачі даних від довжини мережі

Швидкість передачі	Максимальна довжина мережі
1000 Кбіт/сек	40 метрів
500 Кбіт/сек	100 метрів
250 Кбіт/сек	200 метрів
125 Кбіт/сек	500 метрів
10 Кбіт/сек	6 кілометрів

Роз'єми для мережі CAN досі не стандартизовані. Кожен протокол високого рівня зазвичай визначає свій тип роз'ємів для CAN-мережі.

2. Збереження ресурсу автомобіля та завчасне попередження про несправність механізму.

Оскільки алгоритм, що розробляється в даній магістерській дисертації, буде повністю адаптивний, то необхідно власноруч обмежувати алгоритм, щоб

навантаження на агрегати зростали лише в межах ресурсу, що був закладений заводом виробником.

Це є необхідним обмеженням, незважаючи на те, що механізми здатні короткотривало витримати навантаження на 200% більше ніж заводське, довготривале навантаження знищить двигун автомобіля.

В першу чергу буде обмежуватися максимальний тиск повітря, що буде робити турбокомпресор, оскільки пошкодження цього механізму критично для отримання потужності та для функціонування всього двигуна оскільки осколки крильчатки компресуючого колеса турбіни можуть потрапити в циліндри двигуна і знищити його.

Також слід обмежити впорскування палива в циліндри. Це може спричинити завчасне зношування паливних форсунок, паливного насоса високого тиску та паливного насоса низького тиску. Оскільки лямбда-зонд генерує напругу шляхом попадання не спаленого бензину на сам датчик, то збагачена суміш на бензин може спалити лямбда-зонд високою напругою, а у випадку збіднілої паливно-повітряної суміші на бензин лямбда-зонд може згоріти від високої температури випускних газів.

3. Легкість в діагностиці.

Легкість в діагностиці є максимально необхідною вимогою до кожного програмного продукту двигуна автомобіля.

Правильне налаштування програмного забезпечення двигуна починається з правильної діагностики. Більшість станцій технічного обслуговування вимагають повернення автомобіля на заводське програмне забезпечення, оскільки не розуміють як має працює автомобіль після корегування програмного коду.

Враховуючи те, що алгоритм буде лише змінювати заводські задані параметри, а не прописувати свої, то принципи роботи всіх механізмів залишаться без змін. Це повністю зробить діагностику автомобіля після перепрограмування електронного блоку керування такою ж самою, якби двигун автомобіля працював

під керуванням заводського програмного забезпечення. Тому після продажу автомобіля новому власнику не буде створено ніяких складнощів.

4. Підвищення потужності двигуна та зменшення витрати пального.

Не зважаючи на те, що після перепрограмування блоку управління автомобіль стає потужнішим, а для більшої потужності необхідна більша витрата палива та повітря. Оскільки максимальна потужність досягається лише у випадку максимального навантаження, то при спокійній їзді є можливість зменшити витрату пального.

Це можна зробити збільшивши кут запалювання паливно-повітряної суміші, а саму суміш зробити збіднілою на бензин. Температура випускних газів збільшиться і це почистить камеру згорання, циліндри та випускні канали від надлишкового нагару. Підняття температури випускних газів буде компенсуватися раннім кутом запалювання та тим, що двигун буде працювати з низьким навантаженням.

5. Еластичність в подальших налаштуваннях.

У випадку якщо власник автомобіля захоче поставити продуктивніший інтеркулер, якщо буде встановлена приймальна випускна труба без каталізатора або зі спортивним каталізатором, відключити систему AdBlue або систему рециркуляції випускних газів, то заводська програма покращена адаптивним алгоритмом дозволить це зробити. Відключення додаткових систем в даному випадку може бути обмежено лише вміннями інженера, що працюватиме над програмним видаленням цих систем.

Реалізація підвищення потужності до рівня Stage 2 може бути виконана розробленим алгоритмом, для цього буде необхідно власноруч підняти максимально допустимі параметри, що будуть задаватися програмним забезпеченням.

Програмне забезпечення рівня Stage 3 не можливо буде реалізувати за допомогою адаптивного алгоритму, оскільки такі високі програмні налаштування

слід виконувати вже на прикладному рівні. Якщо підняти максимальні значення адаптивний алгоритм зможе працювати, але гарантувати максимальну якість роботи такого програмного забезпечення не можливо, оскільки навіть компанії, які роками розробляють програмне забезпечення не завжди можуть це коректно реалізувати.

4.2 ОПИС РОБОТИ АДАПТИВНОГО АЛГОРИТМУ ПІД ВЕЛИКИМ НАВАНТАЖЕННЯМ

Алгоритм не функціонує впродовж перших 300 кілометрів після інсталяції його в блок управління автомобілем. В цей час відбувається аналізування стилю поведінки водія.

Функціонування алгоритму починається лише після подоланням автомобілем 300 кілометрів та запалювання автомобіля було перезапущено.

Якщо автомобіль рухається під навантаженням, то алгоритм починає множити задані параметри роботи двигуна на певні коефіцієнти.

Наприклад:

Якщо заводський параметр тиску повітря у впускному колекторі 1.1 бар, то щоб досягти максимального тиску в 1.35 бар варто таблицю, що відповідає за тиск у впускному колекторі має множитися на коефіцієнт 1.22.

Важливим показником є не лише тиск повітря, але і кількість повітря. Не зважаючи на те, що кількість повітря залежить від тиску і можливо точно розрахувати яка кількість повітря буде при тому чи іншому тиску. Формула розрахунку кількості повітря:

$$V = \frac{0.25 * Q_c * p_1 * T_o}{f_{\max} * (p_u - p_L) * T_1}$$

V – об'єм повітряного ресіверу, л

Q_c – продуктивність компресору, л/с

p_1 – тиск повітря на вході компресора, бар

T_1 – максимальна температура на вході компресора, К

T_2 – температура стисненого повітря в ресівері, К

$p_u - p_L$ – задана різниця тиску навантаження та завантаження

F_{\max} – максимальна частота

Але є багато факторів, що можуть вплинути на це. На сучасних автомобілях впускні колектори мають можливість змінювати свою довжину в реальному часі та зменшувати кут відкриття дросельної заслонки. Як правило, на автомобілях, що оснащені механічною коробкою передач з ціллю запобігання пошкодження щеплення програм не забезпечення зменшує кут відкриття дроселя, щоб зменшити кількість повітря, що поступає до камери згорання, тим само зменшити крутний момент.

Якщо до зміни програмного забезпечення кількість повітря було 680 міліграм на цикл, є можливість помножити значення на коефіцієнт 1.7, кількість повітря зросте до 1155 міліграм на цикл, тим само крутний момент також зросте в 1.7 раз.

Кількість пального, що впорскується в циліндри може бути як меншою, так і більшою. Все залежить від цілі на яку суміш буде налаштований двигун. Налаштовувати на роботу на бідній суміші можливо, але досить небезпечно та складно, оскільки температура випускних газів буде зростати. Тому не варто давати можливість алгоритму самостійно працювати на збіднілих палим сумішах.

Якщо в циліндрах відбувається непередбачуване запалювання, тобто детонація, то кут запалювання має ставати нижче. При більшому тиску повітря необхідно кут запалювання зробити меншим ніж у заводському програмному забезпеченні, але якщо якість пального досить високе і детонації в циліндрах не відбувається, то кут запалювання можна підняти. Більший кут запалювання також може впливати на більший крутний момент. Для кута запалювання виставляти кути запалювання не можна, оскільки це досить небезпечно. Можна активувати можливість підвищення кута запалювання, якщо нема ніяких негативних впливів.

Як було описано в розділі 2 не всі коефіцієнти мають бути однаковою, але необхідна строга залежність коефіцієнтів між собою. Тобто, кількість повітря має залежати від кількості пального, що впорскується, чим більше буде повітря, тим більше потрібно пального, щоб не зламати двигун. Але не може бути конкретної залежності між кількості пального і кутом запалювання.

4.3 ОПИС РОБОТИ АДАПТИВНОГО АЛГОРИТМУ ПІД МАЛИМ НАВАНТАЖЕННЯМ

Коли не відбувається навантаження на двигун, то є чудова можливість з ціллю економії пального понизити певні параметри, а деякі підняти. За рахунок точного контролю згорання паливно-повітряної суміші можна досягти меншої витрати пального.

При їзді в спокійному режимі можна підняти кут запалювання до великого значення, щоб отримати більший крутний момент і тим самим покращити відгук на педаль акселератора, а також досягти кращого згорання паливно-повітряної суміші в камері згорання, оскільки чим краще відбувається згорання пального – тим менше буде витрата пального.

Без навантаження також можна зменшити кількість пального, що впорскується шляхом множення паливної карти на коефіцієнт менше 1 відносно заводського програмного забезпечення.

Суміш буде збідніла на пальне при спокійній їзді та кут запалювання більше, відповідно температура випускних газів буде зростати, але кількість обертів колінчатого валу двигуна буде меншою, тому температура буде в допущених межах.

Перехід між режимом низького навантаження та повного навантаження може відбутися досить швидко і алгоритм також швидко відреагує на це, оскільки відразу буде запущена карта, що відповідає за роботу двигуна на великому навантаженні.

4.4 РОБОТА З НЕГАТИВНИМИ ПРОЯВАМИ

Робота з негативними проявами необхідна бути і варто на це звертати велику увагу, щоб програмне забезпечення могло врятувати двигун автомобіля від завчасної поломки.

Якщо під великим навантаженням, коли алгоритм продовжує піднімати коефіцієнт відбувається детонація або короткотривала корекція паливно-повітряної суміші і виходить за межі норми або невиконання певним механізмом заданих програмним забезпеченням параметрів, то програма не зменшуватиме коефіцієнт протягом певного часу, щоб виявити чи проблема пов'язана через програмне забезпечення чи через навколишні умови, як температура повітря або якість пального.

Якщо відбувається детонація в камері згорання, то на це є низка причин, таких як висока температура повітря, що втягується, низька якість пального, що впорскується, погана якість свічок або катушок запалювання, паливно-повітряна суміш збідніла на пальне або зависокий. Щоб переконатися, що програмне забезпечення не спричиняє детонацію, то слід певний час продовжувати піднімати кут запалювання, якщо з ростом куту запалювання кути відкатів будуть зменшуватися або не буде чіткої залежності, то причина буде полягати не в програмному забезпеченні, а якщо з ростом куту запалювання буде зростати кут зменшення запалювання, то причина буде полягати саме в програмному забезпеченні.

Якщо негативний прояв буде критичним, то необхідно вивести помилку, з кодом що їй відповідає. Якщо дана помилка вимагає перехід в аварійний режим, то робота алгоритму буде обмежена і алгоритм повернеться до повноцінного функціонування лише після деактивації помилки.

У випадку коли певний механізм не буде виконувати заданих параметрів або перевиконуватиме їх необхідно відобразити помилку з її кодом в списку неполадок та перевести автомобіль в аварійний режим роботи двигуна.

Якщо помилка з'являється в режимі низького навантаження на двигун, то вона пов'язана з несправністю певного механізму.

Такий підхід до сприйняття некоректної роботи двигуна допоможе водію в обслуговуванні транспортного засобу та зменшить ризики зламати двигун автомобіля. Виявлення та перехід до аварійних режимів має відбуватися значно раніше ніж в заводському програмному забезпеченні тому, що автомобіль експлуатується під більшим навантаженням і вимагає більш ретельного обслуговування та догляду.

Висновки до розділу 4

В розділі 4 було розглянуто основні принципи функціонування адаптивного алгоритму, що змінює роботу механізмів, що керуються електронним блоком управління.

Алгоритм базується на 5 основних принципах:

1. Аналіз;
2. Збереження ресурсу автомобіля та завчасне попередження про несправність механізму;
3. Легкість в діагностиці;
4. Підвищення потужності двигуна та зменшення витрати пального;
5. Еластичність в подальших налаштуваннях.

Основаючись на фізичних формулах та на принципах роботи механізмів двигуна було проаналізовано як можна змінити управління механізмів, оскільки вони керуються електронікою.

На прикладі було продемонстровано як можна змінити параметри роботи, щоб отримати кращу потужність під високим навантаженням та зменшити кількість використаного пального при використанні двигуна без навантаження.

5. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ РОБОТИ РОЗРОБЛЕНИХ ЗАСОБІВ, ЩО РЕАЛІЗУЮТЬ АДАПТИВНИЙ АЛГОРИТМ

5.1 РЕЗУЛЬТАТИ РОБОТИ АДАПТИВНОГО АЛГОРИТМУ.

Для тестування ПЗ, що реалізують розроблений алгоритм було обрано автомобіль Volkswagen Golf 5-го покоління з двигуном 1.9 TDI, під управлінням електронного блоку керування BOSCH EDC16U3х/CP.

Для додавання алгоритму до заводського програмного коду управління необхідно спочатку отримати вихідну програму. Це можна зробити підключенням до блоку управління шляхом підпаювання проводів до програматору або приєднатися до діагностичного роз'єму OBD II.

В даному випадку є можливість отримати повне програмне забезпечення лише підключенням до діагностичного роз'єму. Для цього необхідний апаратний засіб KESSv2 виробництва Alientech або CMD Flash. Оскільки у Alientech є свій програмний засіб для модифікації отриманого програмного забезпечення ECM Titanium, то було обрано KESS для отримання коду.

Після додавання розробленого адаптивного алгоритму процес інсталяції готової програми в електронний блок управління автомобіля відбувається також через діагностичний роз'єм.

Процес аналізу даних відбувався впродовж перших 300 кілометрів. На цьому етапі робота двигуна автомобіля нічим не відрізнялася від роботи двигуна на заводському програмному забезпеченні.

Коли алгоритм проаналізував стиль керування автомобілем після наступного запуску двигуна вже були відчутні зміни, а також це було підтверджено в ході діагностики.

При пересуванні по місту змін майже не відбулося. Всі зміни були помітні лише коли оберти двигуна перевищували 2300 обертів на хвилину. Також покращився відгук двигуна на натискання на педаль акселератора. В ході діагностики було помічено, що блок електро-магнітних клапанів почав працювати

значно швидше і відсоток відкриття став дещо меншим, цим само було отримано вищий тиск повітря.

При їзді по трасі зміну були відчутні значно чіткіше. На заводському програмному налаштуванні двигуна автомобіль після швидкості 120 км/год вже не міг похвастатися захопливою динамікою. З алгоритмом ефект слабкої динаміки починається вже після швидкості 170 км/год. Різниця досить велика і вражаюча.

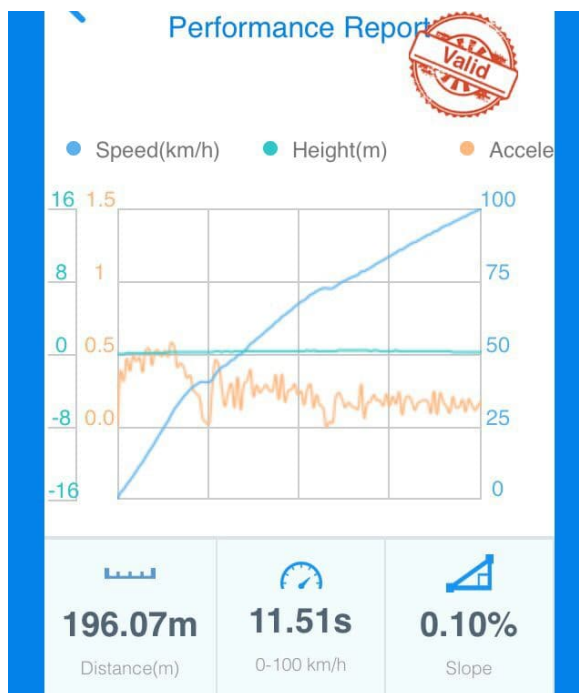


Рисунок 13 – Динаміка 0-100 км/год на заводському програмному налаштуванні

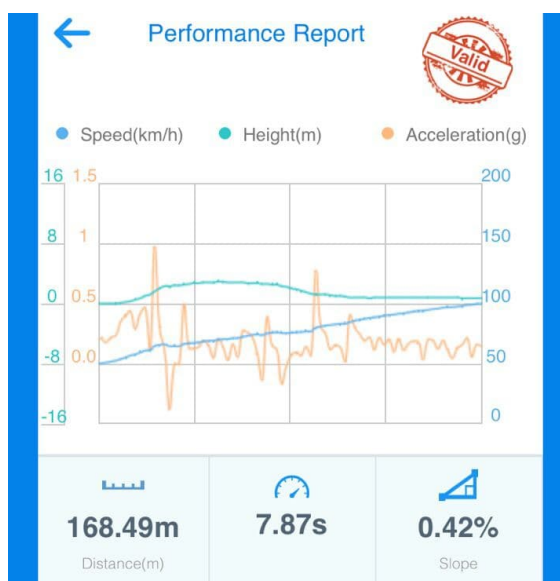


Рисунок 14 – Динаміка 50-100 км/год на заводському програмному налаштуванні

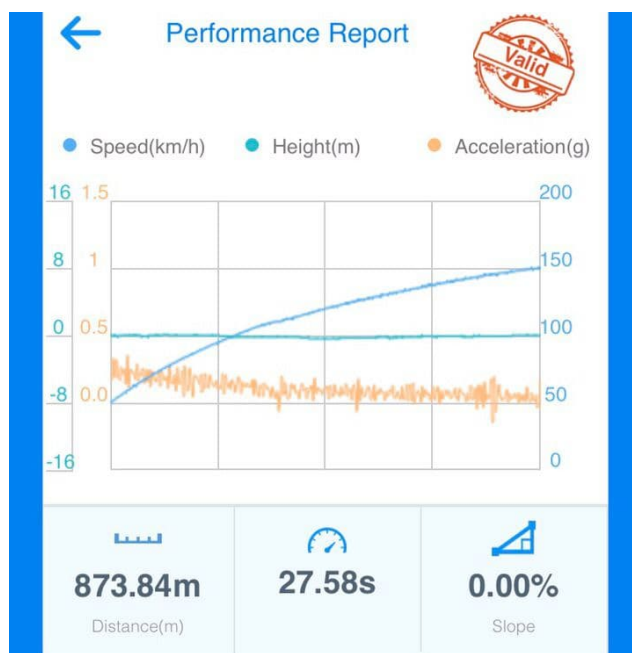


Рисунок 15 – Динаміка 50-150 км/год на заводському програмному налаштуванні

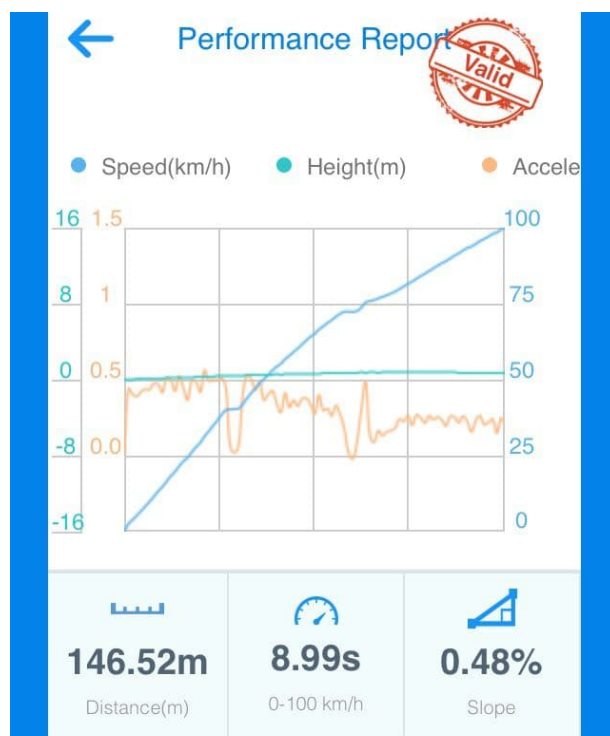


Рисунок 16 – Динаміка 0-100 км/год на покращеному програмному налаштуванні

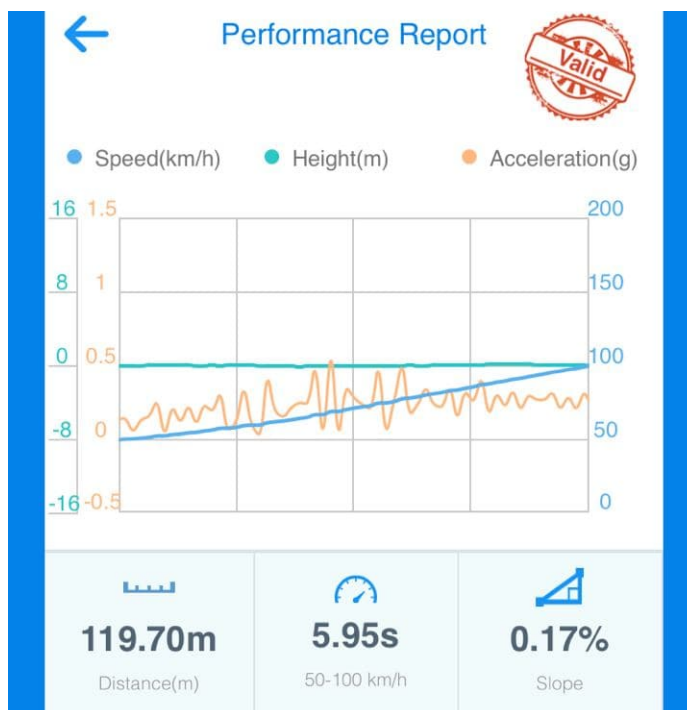


Рисунок 17 – Динаміка 50-100 км/год на покращеному програмному налаштуванні

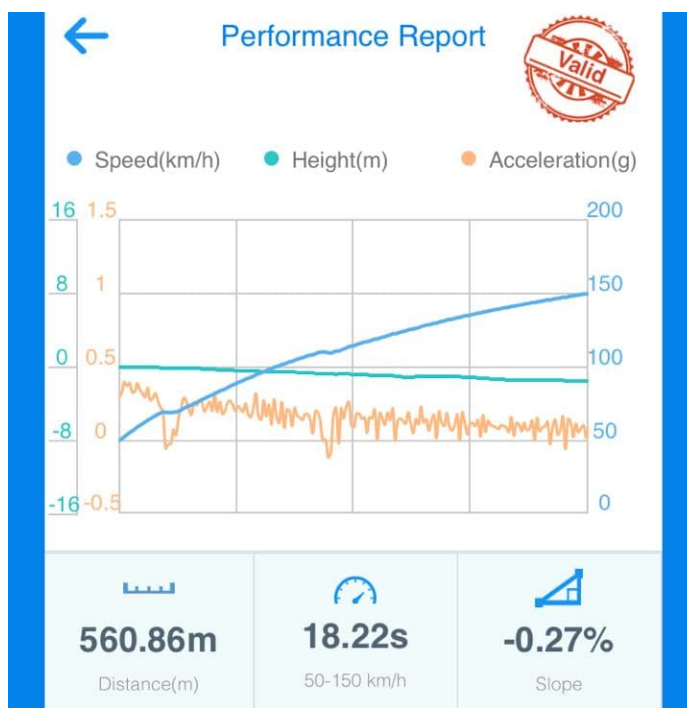


Рисунок 18 – Динаміка 50-150 км/год на покращеному програмному налаштуванні

Як видно по даним замірів динаміки автомобіля час розгону скоротився. Дуже значна різниця в замірі 50-150 км/год обумовлена тим, що діапазон максимального робочого тиску турбіни значно збільшився.

Для порівняння на цей же автомобіль було інстальоване програмне забезпечення Superchips.

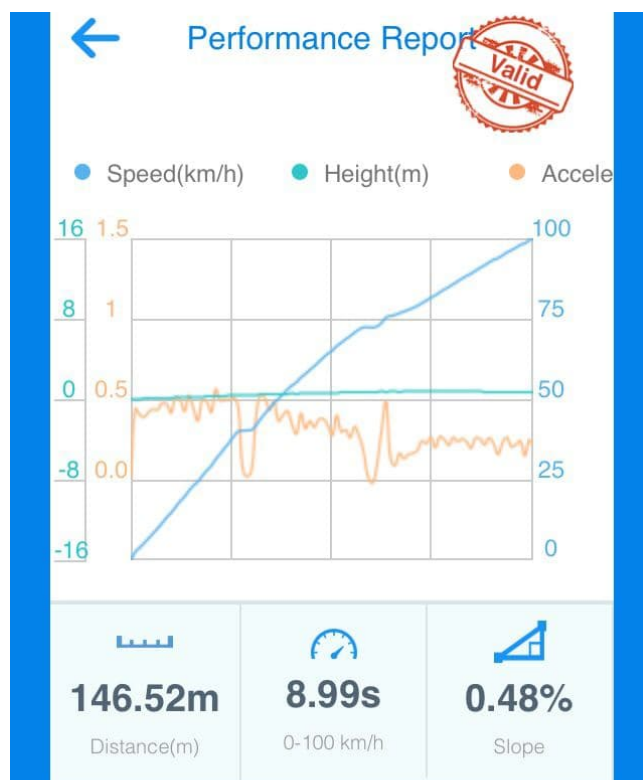


Рисунок 19 – Динаміка 0-100 км/год на програмному налаштуванні Superchips

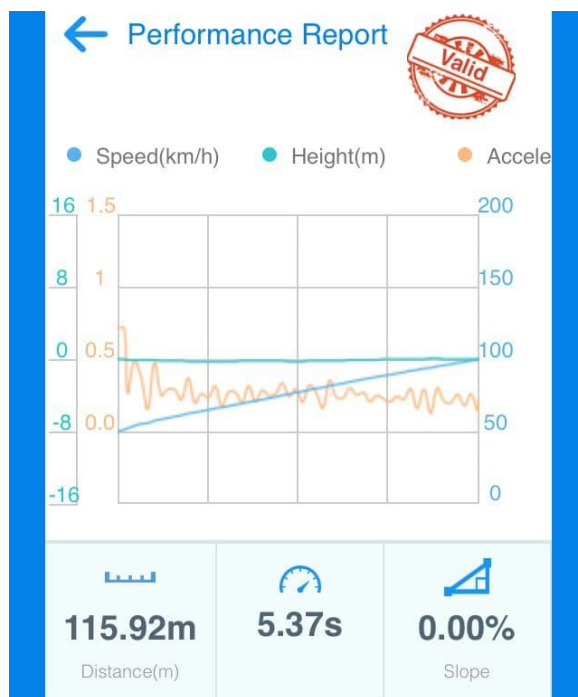


Рисунок 20 – Динаміка 50-100 км/год на програмному налаштуванні Superchips

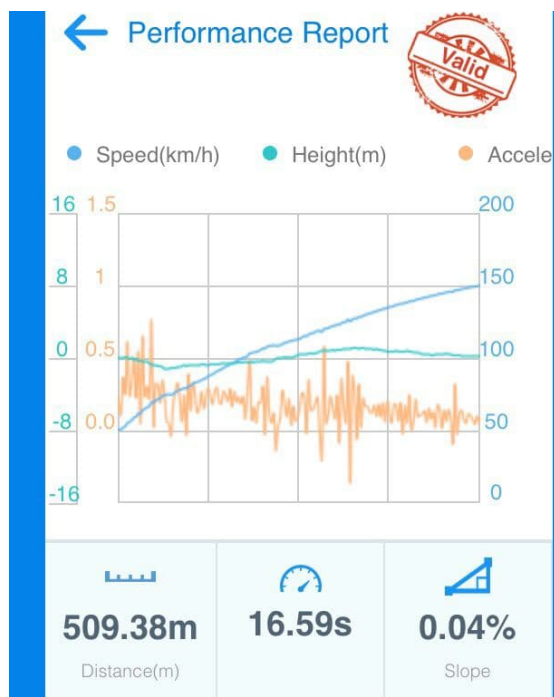


Рисунок 21 – Динаміка 50-150 км/год на програмному налаштуванні Superchips

Основаючиись на отриманих даних помітно, що результати динаміки програмного забезпечення разом з розробленим алгоритмом дещо гірше за

результати динаміки програми розробленої компанією Superchips. Це обумовлено тим, що інженер проектував програмне забезпечення спеціально під автомобіль.

Після подолання 500 км в агресивному режимі алгоритм адаптувався і почав вражати покращеним відгуком на педаль акселератора і максимальний надув став бути доступним вже з 1800 обертів на хвилину.

Але досягти зниження витрати пального при спокійній експлуатації не вдалося досягти. Це обумовлено механічною конструкцією роботи двигуна, тому на двигуну такого типу це зробити не вдасться.

Для перевірки працездатності алгоритму було прийняте рішення інсталиувати програму на ще один автомобіль. Це був Audi Q5 з двигуном 2.0 TDi під управління електронного модулю керування BOSCH EDC17CP20/CP14.

Нажаль, для отримання заводського програмного налаштування необхідне підключення до електронного блоку управління шляхом підпаювання. Оскільки доступ до електронного блоку управління у складно доступному місці, то довелось зробити віртуальне зчитування програмного забезпечення через діагностичний роз'єм, а повне програмне налаштування знайти в мережі інтернет.

Після додавання адаптивного алгоритму до заводського програмного забезпечення програмний продукт було інстальовано через діагностичний роз'єм.

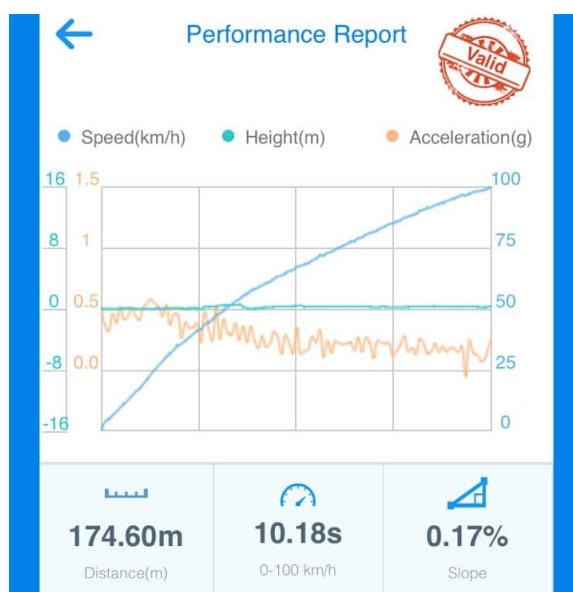


Рисунок 22 – Динаміка 0-100 км/год на заводському програмному налаштуванні



Рисунок 23 – Показники бортового комп'ютера витрати пального на заводському програмному налаштуванні.

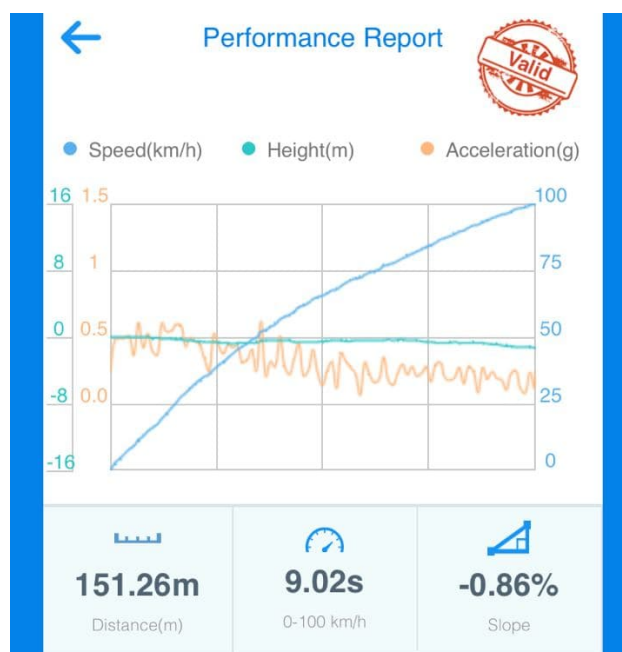


Рисунок 24 – Динаміка 0-100 км/год на покращеному програмному налаштуванні



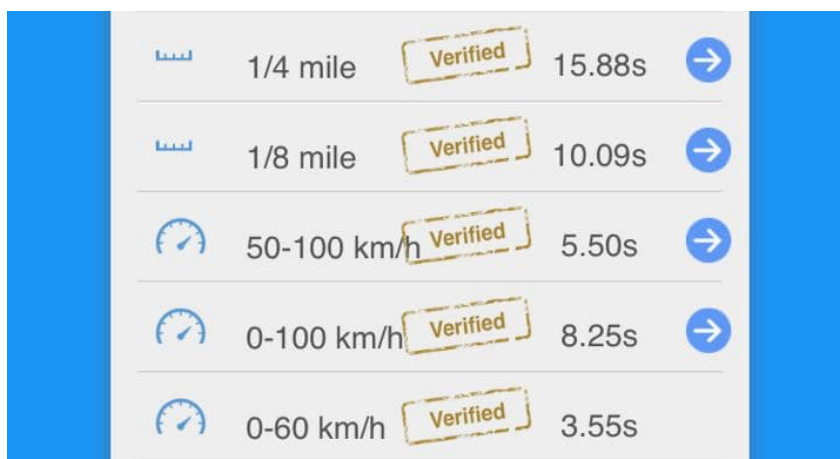
Рисунок 25 – Показники бортового комп'ютера витрати пального на заводському програмному налаштуванні.

По результатам замірів та по показникам бортового комп'ютера помітно, що автомобіль став швидше та економичніше. Продіагностувавши двигун було помітно, що електронне керування турбіни було змінено, а також змінено управління паливними форсунками.

Відразу після того як алгоритм покращив потужність з'явилися негативні прояви роботи. Це проявлялось у надмірній кількості сажі у випускних газах. Двигун такого типу слід форсувати лише більшою кількістю пального, а тиск повітря піднімати можна лише мінімально.

У даному випадку алгоритм підняв тиск турбокомпресора значною мірою і відповідно до того тиску піднялась і кількість пального, що потрапило до циліндру. Щоб не виникало таких проблем довелося обмежити максимальний тиск турбокомпресора.

Після змін в програмному забезпеченні змінились результати замірів динаміки, автомобіль показав значно кращий результат заміру 0-100 км/год та скоротилась витрата пального.








	1/4 mile	Verified	15.88s	→
	1/8 mile	Verified	10.09s	→
	50-100 km/h	Verified	5.50s	→
	0-100 km/h	Verified	8.25s	→
	0-60 km/h	Verified	3.55s	

Рисунок 26 – Динаміка автомобіля після змін в адаптивному алгоритмі

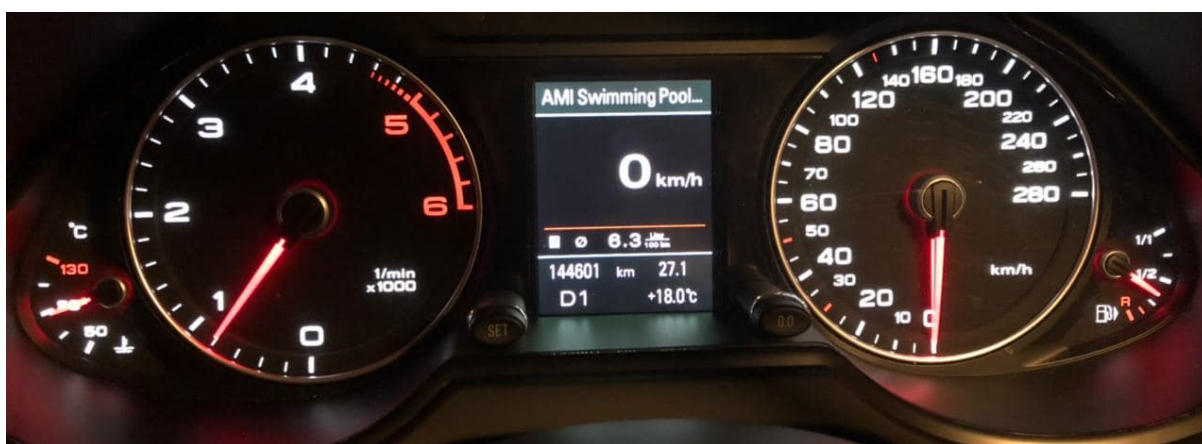


Рисунок 27 – Показники бортового комп'ютера витрати пального після змін в адаптивному алгоритмі.

Висновки до розділу 5

В розділі 5 було проведено тестування роботи адаптивного алгоритму на двох автомобілях. Результати динаміки обох автомобілів стали значно кращими. Але необхідно правильно виставляти обмеження в програмному коді.

Тести відбулися успішно, було досягнуто підняття потужності двигуна автомобіля та досягнута економічність.

В ході тестування виявлено, що при не коректному налаштуванні обмежувачів в програмному коді, адаптивний алгоритм може негативно впливати на роботу двигуна.

Було прийнято рішення змінити обмеження для конкретного автомобіля. Це дало можливість покращити результати роботи двигуна та знизити витрату пального на 12,5%. Потужність вдалось збільшити на 50% відносно заводських налаштувань.

ВИСНОВКИ

В даній магістерській дисертації було розроблено адаптивний алгоритм керування роботою двигуна. Такий алгоритм необхідний для покращення потужнісних, динамічних та економічних характеристик автомобіля.

Проведений роботі порівняльний огляд показав, що для ефективного програмного засобу необхідно мати інформацію не лише про можливості механізмів для яких буде змінено керування, а й інформацію про параметри, що впливають на ресурс, який був закладений заводом виробником.

Також, розглянуто такі способи покращення характеристик потужності двигуна, як:

- підвищення потужності коефіцієнтами;
- підвищення потужності шляхом описування моделі;
- підвищення потужності лише з одним параметром;
- індивідуальне налаштування на основі показників лямбда-зонду.

В ході аналізу було вирішено яким способом буде розроблятися адаптивний алгоритм, а саме шляхом множення заводських параметрів на коефіцієнти, кожний параметр повинен мати свій особливий коефіцієнт та буде мати можливість змінюватися у реальному часі в залежності від стилю керування автомобілем.

Розглянуто та розроблено основні принципи на яких має базуватися адаптивний алгоритм, а саме:

1. Аналіз роботи двигуна автомобіля;
2. Збереження ресурсу автомобіля та завчасне попередження про несправність механізму;
3. Легкість в діагностиці;
4. Підвищення потужності двигуна та зменшення витрати пального;
5. Еластичність в подальших налаштуваннях.

Основою на фізичних формулах та на принципах роботи механізмів двигуна було показано як можна змінити управління механізмів, оскільки вони керуються електронікою.

Проведено дослідні тести роботи адаптивного алгоритму на реальних автомобілях.

Тести відбулися успішно, було досягнуто підняття потужності двигуна автомобіля та досягнута економічність.

В ході тестування було виявлено, що при не коректному налаштуванні обмежувачів в програмному коді, адаптивний алгоритм може негативно впливати на роботу двигуна.

В результаті тестування було прийнято змінити обмеження для конкретного автомобіля, що дало можливість покращити результати роботи двигуна та знизити витрату пального. Економічність автомобіля зросла майже на 15%, а потужність зросла на 50% у порівнянні із заводською.

Мета магістерської дисертації була досягнута. А саме підвищення ефективності (збільшення потужності та зменшення витрат палива автомобілем) за рахунок створення нових програмно-апаратних засобів для реалізації адаптивного алгоритму керування двигуном автомобіля і, робота яких, буде заснована на аналізі характеристик навколишнього середовища та внутрішніх параметрів двигуна, трансмісії й інших вузлів автомобіля.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. A E Hergenhan, G Heiser. Operating systems technology for converged ECUs[C]. 2008
2. H.Heinecke. Automotive system design - challenges and potential. Design, Automation and Test in Europe. 2005:656~657
3. Jurgen Mossinger. Software in Automotive Systems. IEEE Software, March/April 2010. IEEE Computer Society, 2010.
4. P Hladik, A Deplanche, S E B Faucou, et al. Adequacy between AUTOSAR OS specification and real-time scheduling theory[C]. 2007: 225-233.
5. AUTOSAR Development Partnership. AUTOSAR – Current results and preparations for exploitation. 7th EUROFORUM conference 'Software in the vehicle', Stuttgart 2006
6. S Furst. AUTOSAR Newsletter Q2/2010[Z]. AUTOSAR, 2010.
7. K Klobedanz, C Kuznik, A Thuy, et al. Timing modeling and analysis for AUTOSAR-based software development: a case study[C]. 2010: 642-645.
8. Caliebe P, Lauer C, German R. Flexible integration testing of automotive ECUs by combining AUTOSAR and XCP[C]. 0004-07- 20, 2011, (67-72).Computer Applications and Industrial Electronics (ICCAIE), 2011 IEEE International Conference on.
9. T Scharnhorst, H Heinecke, K Schnelle, et al. Application to engine control at Renault AUTOSAR-Challenges and Achievements 2005[J]. VDI BERICHTE, 2005, 1907: 395.
10. K Lakshmanan, G Bhatia, R Rajkumar. Integrated end-to-end timing analysis of networked autosar-compliant systems[C]. 2010: 331-334.
11. Idrees M S, Schweppe H, Roudier Y, et al. 2011 Secure automotive on-board protocols: A case of over-the-air firmware updates[M] Communication Technologies for Vehicles 224-238
12. Ye B F 2010 Research on Android operating system transplantation and key technologies[D] Guangzhou: Ji'nan University

13. Wu W F 2015 Research and realization of the security technology of electricity bill based on U shield technology[D] Beijing: North China electric power university
14. Vanholme B, Lusetti B, Gruyer D, et al. Highly automated driving on highways: System implementation on PC and automotive ECUs[C]. 0005-07-20, 2011, (1465-1470).Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on.
15. Liu H 2013 Research on the self-organizing network information authentication and privacy protection mechanism[D] Xi'an: Xi 'an university of electronic science and technology
16. Senthilkumar K, Ramadoss R. Designing multicore ECU architecture in vehicle networks using AUTOSAR[C]. 2011, (270-275).Advanced Computing (ICoAC), 2011 Third International Conference on.
17. Phung P H and Nilsson D K 2010 A model for safe and secure execution of downloaded vehicle applications[C] Road Transport Information and Control Conference and the ITS United Kingdom Members' Conference (RTIC 2010) - Better transport through technology 1-6